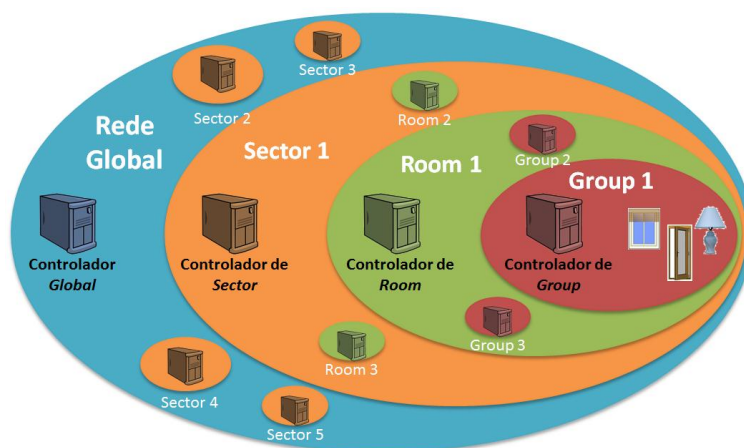




VASCO JOSÉ
DIAS BAPTISTA

B-LIVE WIRELESS: UM PROTOCOLO DE TEMPO- REAL PARA DOMÓTICA HABITACIONAL

B-LIVE WIRELESS: A REAL-TIME PROTOCOL FOR HOME AUTOMATION





**VASCO JOSÉ
DIAS BAPTISTA**

**B-LIVE WIRELESS: UM PROTOCOLO DE TEMPO-
REAL PARA DOMÓTICA HABITACIONAL**

**B-LIVE WIRELESS: A REAL-TIME PROTOCOL FOR
HOME AUTOMATION**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Doutor José Alberto Gouveia Fonseca, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro



Fundo Europeu de Desenvolvimento Regional



Aos meus pais, à minha família, à minha namorada, a todos os meus amigos e professores

o júri

presidente

Prof. Doutor Alexandre Manuel Moutela Nunes Da Mota

professor associado do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro

Prof. Doutor Paulo José Lopes Machado Portugal

professor auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor José Alberto Gouveia Fonseca

professor associado do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro

Mestre Paulo Jorge de Campos Bartolomeu

professor assistente da Escola Superior de Tecnologia e Gestão de Águeda, Universidade de Aveiro

agradecimentos

Não é possível esquecer todos aqueles que de uma forma directa ou indirecta estiveram envolvidos na realização deste projecto. A todos os meus mais sinceros e profundos agradecimentos.

Ao meu orientador, Prof. José Alberto Fonseca e ao meu co-orientador, Mestre Paulo Bartolomeu pela grande ajuda prestada, pela confiança em mim depositada, por todos os erros enumerados, pelo incentivo moral e técnico e paciência que permitiu que melhorasse todos os dias o meu desempenho no decorrer do projecto.

Aos meus pais talvez o voto de agradecimento mais profundo, por todos os sacrifícios que fizeram para me permitirem chegar onde cheguei. Por me darem todas as oportunidades que deram e por depositarem em mim toda a confiança possível. A eles devo tudo quanto alcancei. Obrigado por tudo Mãe e Pai e espero que no futuro consiga manter-vos sempre orgulhosos.

A todos os meus amigos que chegaram aqui comigo, aos que foram para longe mas continuam perto, a todos os que fizeram parte do meu ERASMUS na Polónia. Aos que com a mais simples ideia de quem está fora ajudam sem terem noção do que ajudaram. Aos que directamente tentaram ajudar nos momentos mais negros e finalmente a todos os amigos de jantares e convívio, que por qualquer momento nos fazem sentir bem e nos oferecem momentos para recordar e crescer.

À minha namorada que me ajudou e apoiou em todos os momentos bons ou menos bons, e por simplesmente estar presente a meu lado sempre que possível, ainda que tão grande distância nos separe.

Muito obrigado por toda a paciência que tiveram comigo.
Thank you very much for all the patience you had for me.
Dziękuję bardzo za cierpliwość, którą mieliście dla mnie.

Por último, mas não menos importante, gostaria de agradecer à FCT – Fundação para a Ciência e a Tecnologia, ao projecto DHT-Mesh e ao projecto LUL – Living Usability Lab, pela oportunidade e todo o apoio que me deram de forma a permitir que trabalhasse numa área que sempre me fascinou.

palavras-chave

Protocolo, camada protocolar, microcontrolador, comunicação sem fios, rede, IEEE802.15.4, WFTT, bandjacking, sensor, actuador.

resumo

A tendência de envelhecimento da população tem-se vindo a agravar causando uma pressão significativa ao nível das organizações que suportam os serviços de saúde e de acção social. Uma solução para este problema consiste em permitir que a população idosa ou com limitações funcionais possa continuar a viver no conforto das suas habitações. Neste cenário, a habitação terá de integrar um sistema de automação “inteligente” que, além de ajudar as pessoas nas suas actividades diárias, monitorize a sua condição de saúde.

O projecto *Living Usability Lab* tem como objectivo o uso de interfaces multimodais, redes de sensores sem-fios, redes de nova geração e computação distribuída para melhorar a qualidade de vida da população idosa nas suas habitações. No âmbito deste projecto será desenvolvida uma rede heterogénea de sensores e actuadores que permitirá monitorizar o ambiente, a ocorrência de eventos, a localização e movimento de pessoas, e a sua saúde. A Micro I/O é o parceiro do projecto que desenvolve a infra-estrutura de comunicação sem-fios local, detendo *know-how* em domótica devido à experiência anterior no desenvolvimento de uma solução chamada B-Live.

Esta dissertação é enquadrada no desenvolvimento de um protocolo de comunicações que permite não só potenciar as funcionalidades originais do B-Live, utilizando comunicações sem-fios, como também satisfazer os requisitos da rede de sensores e actuadores a desenvolver no âmbito do projecto *Living Usability Lab*. Assim, esta dissertação apresenta um protocolo sem fios capaz de suportar fluxos de tráfego com requisitos de qualidade de serviço diferenciados, mesmo em ambientes abertos em que outras tecnologias de contenção competem pelo meio.

keywords

Protocol, protocol layer, microcontroller, wireless communication, network, IEEE802.15.4, WFTT, bandjacking, sensor, actuator.

abstract

The trend of the elderly population aging has been increasing, putting a significant pressure on the organizations supporting health care and social services. One solution for this problem consists in allowing the elderly and the impaired population to continue living in the comfort of their homes. In this scenario, the home will integrate an “intelligent” automation system that, besides helping them in their daily activities, will monitor their health condition and trigger an alarm whenever an abnormal condition is detected.

The Living Usability Lab project has the goal of using multimodal interfaces, wireless sensor networks, new generation networks and distributed computing to improve the quality of life of the elderly population in their homes. In the scope of this project, a heterogeneous network of sensors and actuators will be developed, allowing to monitor the environment, to detect the occurrence of events, to perform the localization and movement of people as well as to watch their health. Micro I/O is the project partner responsible for developing the local wireless communication infrastructure, having know-how on the development of home automation systems due to previous experience in the B-Live project, a solution for domotics.

This thesis results from the development of a protocol that, besides allowing fueling the original features of B-Live with wireless communications, can satisfy the requirements of the sensor and actuator network to be developed under the scope of the Living Usability Lab project. As such, it presents a wireless communication protocol capable of supporting data flows with differentiated QoS requirements, even in open environments, where other contention technologies may compete for the medium.

Índice

1. Introdução	1
1.1 Objectivos.....	2
1.2 Contribuições	2
1.3 Estrutura da dissertação	3
2. Trabalho Anterior.....	5
2.1 Introdução	5
2.2 Sistemas de automação habitacional sem fios.....	5
2.3 B-Live	10
3. Protocolo B-Live Wireless.....	13
3.1 Serviços oferecidos	13
3.2 Requisitos	14
3.3 Protocolo WFTT	15
3.4 Solução	17
3.4.1 Arquitectura	17
3.4.2 Operação	19
3.5 Elementos do sistema	19
3.6 Modelo de comunicação	22
3.6.1 Características.....	23
3.6.2 Estrutura elementar de uma trama	24
3.6.3 Endereçamento	25
3.6.4 Tipos de tramas	32
3.6.5 Funcionamento	41
4. Desenvolvimento	51
4.1 <i>Software</i>	51
4.1.1 Introdução.....	51
4.1.2 Bibliotecas.....	52
4.1.3 B-Live.....	54
4.2 <i>Hardware</i>	56
4.2.1 Introdução.....	56

4.2.2 Controlo.....	57
4.2.3 Potência.....	58
5. Conclusão e trabalho futuro	63
Bibliografia	65
Anexos.....	67
Anexo 1: Estrutura de um pacote MAC IEEE802.15.4.....	69
Anexo 2: Subgrupos associados ao campo <i>SRS/DST Room</i>	71
Anexo 3: Subgrupos associados ao campo <i>SRS/DST Group</i>	73
Anexo 4: Valores atribuídos a campos da trama B-Live Wireless	81
Anexo 5: Tutorial de configuração do ambiente de desenvolvimento	83
Anexo 6: Esquema da placa de testes	103
Anexo 7: Terminal de ligação da placa de controlo	105

Lista de Figuras

Figura 1: Trama do protocolo B-Live implementado em CAN	10
Figura 2: Trama do protocolo B-Live implementado em RS232.....	10
Figura 3: Interrupção de transmissões devido a um <i>black-burst</i>	15
Figura 4: Transmissão de dados segundo o protocolo WFTT	16
Figura 5: As diferentes partes constituintes de uma trama WFTT	16
Figura 6: Arquitectura da rede B-Live Wireless	18
Figura 7: Processo de registo e análise de eventos na rede	19
Figura 8: Os actuadores controlam outros dispositivos da habitação.....	20
Figura 9: Os sensores obtêm informações sobre o ambiente controlado	20
Figura 10: O controlador interliga os sensores e actuadores ao <i>gateway</i>	21
Figura 11: O <i>gateway</i> cria uma ponte entre a WSN e o servidor	21
Figura 12: Estrutura de ligações entre as diferentes categorias de dispositivos	22
Figura 13: Facilidade de adaptação do protocolo B-Live Wireless a outros protocolos	23
Figura 14: Estrutura de uma trama genérica.....	24
Figura 15: Campos de endereçamento de origem.....	25
Figura 16: Campos de endereçamento de destino.....	28
Figura 17: Diferença e localização de controladores globais, de sector, de divisão e grupo	31
Figura 18: Os campos variáveis dependem do tipo de trama	32
Figura 19: Estrutura de uma trama de ordem de operação	32
Figura 20: Exemplo de uma trama de ordem de operação normal	33
Figura 21: Estrutura de uma trama de <i>acknowledge</i>	33
Figura 22: Exemplo de uma trama de <i>acknowledge</i>	33
Figura 23: Estrutura de uma trama de tempo-real.....	33
Figura 24: Exemplo de uma trama de tempo-real	34
Figura 25: Variação do nível de RSSI com a distância ao emissor	34
Figura 26: Estrutura de uma trama de localização	34
Figura 27: Exemplo da operação de localização.....	35
Figura 28: Estrutura de uma trama de pedido " <i>Are you alive?</i> "	35
Figura 29: Exemplo da operação de pedido " <i>Are you alive?</i> "	35
Figura 30: Estrutura de uma trama de informação " <i>I'm alive!</i> "	36
Figura 31: Exemplo da utilização da trama " <i>I'm alive!</i> "	36
Figura 32: Estrutura de uma trama de pedido de <i>status</i>	36
Figura 33: Exemplo da utilização da trama de pedido de status	36

Figura 34: Estrutura de uma trama de informação de <i>status</i>	37
Figura 35: Exemplo da utilização da trama de informação de <i>status</i>	37
Figura 36: Estrutura de uma trama de erro	37
Figura 37: Exemplo da utilização da trama de erro por outro dispositivo	38
Figura 38: Exemplo da utilização da trama de erro pelo próprio dispositivo	38
Figura 39: Estrutura de uma trama de configuração	38
Figura 40: Exemplo da operação de configuração.....	39
Figura 41: Estrutura de uma trama de configuração	39
Figura 42: Exemplo de uma trama de <i>alteração de endereços</i>	40
Figura 43: Estrutura de uma trama de <i>acknowledge de configuração de dispositivo</i>	40
Figura 44: Estrutura de uma trama de <i>acknowledge de alteração de endereços</i>	40
Figura 45: Exemplo de uma trama de <i>acknowledge de configuração</i>	41
Figura 46: Diferentes tipos possíveis de programadores	41
Figura 47: Exemplo de informação no programador portátil.....	42
Figura 48: Configuração de um dispositivo - passo I	42
Figura 49: Configuração de um dispositivo - passo II.....	42
Figura 50: Configuração de um dispositivo - passo III.....	42
Figura 51: Configuração de um dispositivo – completa	42
Figura 52: Um exemplo de um cenário onde o sistema se encontra aplicado.....	43
Figura 53: Exemplo de controlo de iluminação – passo I	44
Figura 54: Exemplo de controlo de iluminação – passo II	44
Figura 55: Exemplo de controlo de iluminação – passo III	44
Figura 56: Exemplo de controlo de iluminação – passo IV	44
Figura 57: Exemplo de controlo de iluminação – passo V.....	44
Figura 58: Exemplo de controlo de iluminação – passo VI	44
Figura 59: Exemplo de localização – passo I.....	45
Figura 60: Exemplo de localização – passo II.....	45
Figura 61: Exemplo de localização – passo III.....	45
Figura 62: Exemplo de localização – passo IV	45
Figura 63: Exemplo de localização – passo V	45
Figura 64: Exemplo de detecção de eventos – passo I	46
Figura 65: Exemplo de detecção de eventos – passo II	46
Figura 66: Exemplo de detecção de presença – passo I	46
Figura 67: Exemplo de detecção de presença – passo II	46
Figura 68: Exemplo de detecção de alerta e alarme	47
Figura 69: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo I	47
Figura 70: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo II	48
Figura 71: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo III	48

Figura 72: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo IV	48
Figura 73: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo V	49
Figura 74: Funcionamento global do sistema com o protocolo <i>WFTT</i> – passo VI.....	49
Figura 75: O logótipo da Microchip® e do MPLABX®	51
Figura 76: Esquema de bibliotecas existentes no B-Live Wireless	52
Figura 77: Placa de desenvolvimento utilizada	56
Figura 78: Diagrama de blocos da placa de controlo	57
Figura 79: Simulação da organização possível para as duas faces da placa de controlo	58
Figura 80: Diagrama de blocos da placa de potência de saída	59
Figura 81: Diagrama de blocos da placa de potência de entrada	59
Figura 82: Diagrama de blocos da placa de potência de sensores	60
Figura 83: Simulação da organização possível para a placa de potência de saída	60
Figura 84: Técnica utilizada para detecção automática de funcionalidade	61
Figura 85: Formato do pacote do <i>standard</i> IEEE802.15.4	69
Figura 86: Definições associadas ao campo Frame Control da Figura 85	69
Figura 87: Elementos necessários à instalação do MPLAB X	83
Figura 88: Pacotes do <i>openjdk</i> necessários para o MPLAB X	84
Figura 89: Repositórios necessários para o MPLAB X.....	84
Figura 90: Instalação de pacotes necessários para o MPLAB X	85
Figura 91: Comandos de consola necessários para executar a instalação do MPLAB X	85
Figura 92: Instalação do IDE MPLAB X.....	86
Figura 93: Instalação dos compiladores do IDE MPLAB X.....	86
Figura 94: Pasta dos ficheiros de documentação do compilador C18	88
Figura 95: Documentação dos <i>configuration bits</i> para o PIC utilizado.....	89
Figura 96: <i>Configuration bits</i> introduzidos no código.....	90
Figura 97: Aplicação para configurar os <i>drivers</i> dos programadores	92
Figura 98: Aplicação para configurar os <i>drivers</i> dos programadores	93
Figura 99: Janela de configuração do projecto	94
Figura 100: Janela de configuração do ICD	94
Figura 101: Botões de operação do ICD no MPLAB X.....	95
Figura 102: Ficheiro de instalação existente no <i>website</i>	96
Figura 103: Pasta com os ficheiros extraídos referentes ao ds30Loader	96
Figura 104: Repositórios necessários para a instalação do Mono	97
Figura 105: Pasta com os ficheiros extraídos referentes ao ds30Loader	98
Figura 106: O dispositivo USB-SERIE é mostrado na lista de <i>devices</i>	99
Figura 107: Janela de interface do <i>ds30 Loader</i>	99

Lista de Tabelas

Tabela 1: Características do protocolo EnOcean	6
Tabela 2: Características do protocolo Insteon	6
Tabela 3: Características do protocolo ONE-NET	7
Tabela 4: Características do protocolo Z-Wave	7
Tabela 5: Características do protocolo ZigBee	8
Tabela 6: Descrição dos valores associados ao campo <i>Bat level</i>	25
Tabela 7: Descrição dos valores associados ao campo <i>SRC Sector</i>	26
Tabela 8: Descrição dos valores associados ao campo <i>SRC Room</i>	26
Tabela 9: Descrição dos valores associados ao campo <i>SRC Group</i>	27
Tabela 10: Descrição dos valores associados ao campo <i>SRC ID</i>	28
Tabela 11: Descrição dos valores associados ao campo <i>DST Sector</i>	29
Tabela 12: Descrição dos valores associados ao campo <i>DST Room</i>	29
Tabela 13: Descrição dos valores associados ao campo <i>DST Group</i>	30
Tabela 14: Descrição dos valores associados ao campo <i>DST ID</i>	30
Tabela 15: Exemplo de configuração dos controladores da Figura 17	31
Tabela 16: Valores associados ao campo <i>Oper.</i> da trama de alteração de endereços	40
Tabela 17: As bibliotecas existentes no B-Live Wireless	54
Tabela 18: Divisões disponíveis para atribuir ao campo <i>SRC/DST Room</i>	71
Tabela 19: Grupos disponíveis para atribuir ao campo <i>SRC/DST Group</i>	73
Tabela 20: Subgrupo de iluminação do campo <i>SRC/DST Group</i>	74
Tabela 21: Subgrupo de portas e janelas do campo <i>SRC/DST Group</i>	74
Tabela 22: Subgrupo de estores do campo <i>SRC/DST Group</i>	75
Tabela 23: Subgrupo de tomadas de potência do campo <i>SRC/DST Group</i>	75
Tabela 24: Subgrupo de tomadas comuns do campo <i>SRC/DST Group</i>	75
Tabela 25: Subgrupo de autoclismo do campo <i>SRC/DST Group</i>	76
Tabela 26: Subgrupo de torneiras do campo <i>SRC/DST Group</i>	76
Tabela 27: Subgrupo de menu vocal do campo <i>SRC/DST Group</i>	76
Tabela 28: Subgrupo de intercomunicador do campo <i>SRC/DST Group</i>	77
Tabela 29: Subgrupo de multimédia do campo <i>SRC/DST Group</i>	77
Tabela 30: Subgrupo de dispensadores do campo <i>SRC/DST Group</i>	77
Tabela 31: Subgrupo de ventilação e aquecimento do campo <i>SRC/DST Group</i>	78
Tabela 32: Subgrupo de interruptores do campo <i>SRC/DST Group</i>	78
Tabela 33: Subgrupo de dispositivos móveis do campo <i>SRC/DST Group</i>	78

Tabela 34: Subgrupo de elevador do campo <i>SRC/DST Group</i>	79
Tabela 35: Subgrupo de sensores do campo <i>SRC/DST Group</i>	79
Tabela 36: Valores associados ao campo <i>OUT</i> da trama de ordem de operação normal	81
Tabela 37: Descrição das causas de erro da trama de diagnóstico de erro.....	81
Tabela 38: Valores associados ao campo <i>TX Power</i> da trama de configuração	82
Tabela 39: Sinais existentes no conector da placa de controlo.....	105

Abreviaturas e acrónimos

BAT	-	<i>Battery</i>
bps	-	<i>Bits per second</i>
CSMA/CA	-	<i>Carrier Sence Multiple Access with Collision Avoidance</i>
EEPROM	-	<i>Electrically Erasable Programmable Read-Only Memmory</i>
FTT / (WFTT)	-	<i>Flexible Time Triggered (Wireless FTT)</i>
GND	-	<i>Ground</i>
ICD2	-	<i>In-Circuit Debugger 2</i>
ICD3	-	<i>In-Circuit Debugger 3</i>
IDE	-	<i>Integrated Development Environment</i>
IEEE	-	<i>Institute of Electrical and Electronics Engineers</i>
I/O	-	<i>Input/Output</i>
ISM band	-	<i>Industrial, Scientific and Medical radio band</i>
Kbps	-	<i>Kilobits per second</i>
KHz	-	<i>Kilo Hertz</i>
LAN	-	<i>Local Area Network</i>
MAC	-	<i>Media Access Control</i>
MHz	-	<i>Mega Hertz</i>
MIPS	-	<i>Millions of Instructions Per Second</i>
MRF	-	<i>Abreviatura para o modulo MRF24J40</i>
ms	-	<i>Milissegundo</i>
P2P	-	<i>Peer-To-Peer</i>
PC	-	<i>Personal Computer</i>
PCB	-	<i>Printed Circuit Board</i>
PIC	-	<i>Programmable Interface Controller</i>
PLC	-	<i>Power Line Communications</i>
PW	-	<i>Protected Window</i>
RAM	-	<i>Random Access Memmory</i>
RISC	-	<i>Reduced Instruction Set Computer</i>
ROM	-	<i>Read-Only Memmory</i>
RS232	-	<i>Recommended Standard 232</i>
RSI	-	<i>Rotina de Serviço à Interrupção</i>
SPI	-	<i>Serial Peripheral Interface</i>
seg	-	<i>Segundos</i>
TDMA	-	<i>Time Division Multiple Access</i>
UART	-	<i>Universal Asynchronous Receiver and Transmitter</i>

URL	-	<i>Uniform Resource Locator</i>
us	-	<i>Microsegundo</i>
VCC	-	<i>Common Colector Voltage</i>
WFTT	-	<i>Wireless Flexible Time-Triggered</i>
Wi-Fi	-	<i>Wireless Fidelity</i>
WLAN	-	<i>Wireless Local Area Network</i>
WPAN	-	<i>Wireless Personal Area Network</i>
WSAN	-	<i>Wireless Sensor/Actuator Network</i>
XTEA	-	<i>Extended Tiny Encryption Algorithm</i>

1. Introdução

Nas últimas décadas tem-se verificado uma tendência de envelhecimento generalizado da população. Os grupos adultos têm-se tornado mais saudáveis, mais formados e com níveis de rendimento superiores [1]. A melhoria dos serviços de saúde e os avanços tecnológicos têm contribuído para aumentar a esperança de vida da população. Estas tendências têm várias consequências [2]: um número maior de pessoas mais idosas quer em termos relativos quer em termos absolutos; um número considerável de idosos com “idade mais avançada”; um número cada vez menor de profissões informais (i.e., “donas de casa”), e uma força de trabalho cada vez menor para contribuir para o financiamento dos serviços sociais e de saúde. Neste sentido, o aumento da população idosa coloca uma forte pressão ao nível dos custos nas organizações que suportam os serviços de saúde e acção social.

Uma solução para o problema descrito consiste em possibilitar que a população idosa ou com limitações funcionais possa continuar a viver no conforto das suas habitações [3], sendo assistida por sistemas inteligentes que a ajudam nas actividades diárias e que monitorizam a sua condição de saúde através da medição de sinais biológicos [4] ou por reconhecimento do comportamento, por exemplo. A área de trabalho das habitações inteligentes com ênfase na saúde é um tópico de investigação importante que tem recebido atenção de um número significativo de projectos, nomeadamente o projecto *Living Usability Lab*, o projecto *MonAMI*, a *Aware Home* e o programa *TAFETA*, para mencionar alguns.

O *Living Usability Lab for Next Generation Networks* (LUL) [5] é um projecto de Investigação e Desenvolvimento participado tanto por empresas (Microsoft, Micro I/O e Plux) como por entidades do Sistema Científico e Tecnológico (Instituto de Engenharia Electrónica e Telemática de Aveiro e Escola Superior de Saúde da Universidade de Aveiro e INESC Porto). O projecto foca-se no desenvolvimento de tecnologias e serviços para suportar cidadãos activos, produtivos e saudáveis. Tem como objectivo o uso de interfaces multimodais (que empregam o princípio do design universal com reconhecimento de discurso e de gestos), redes de sensores sem-fios, redes de nova geração e computação distribuída para melhorar a qualidade de vida da população idosa nas suas habitações.

O projecto LUL tem como meta a criação de um laboratório vivo piloto para avaliar as tecnologias desenvolvidas com utilizadores reais. Este piloto irá incorporar diversas tecnologias para assegurar que os idosos estejam seguros, ligados a familiares, cuidadores e amigos em casa. Um dos componentes importantes do piloto será uma rede heterogénea de sensores e actuadores, que permitirá monitorizar o ambiente (por exemplo, luminosidade), a ocorrência de eventos (por exemplo, alguém se sentou ou levantou), a localização e movimento de pessoa e a sua saúde (por exemplo, medindo a pulsação).

A Micro I/O no projecto LUL, responsável pelo desenvolvimento da rede de comunicações local sem-fios de suporte aos vários sensores e actuadores previstos. Neste âmbito, a empresa detém *know-how* significativo resultante do desenvolvimento de uma solução para domótica habitacional chamada B-Live. Este sistema dirige-se a pessoas com limitações funcionais, sendo suportado numa rede de comunicação cablada baseada na tecnologia *Controller Area Network*. Este projecto foi premiado em 2007 com o prémio Engº Jaime Filipe

que contempla a melhor concepção inovadora e promotora de autonomia, no âmbito da engenharia de reabilitação (atribuído por Instituto da Segurança Social). O presente trabalho é enquadrado no desenvolvimento de um protocolo de comunicações que permita não só potenciar as funcionalidades originais do B-Live utilizando comunicações sem-fios como também satisfazer os requisitos da rede de sensores e actuadores a desenvolver no âmbito do Living Lab.

1.1 Objectivos

Nesta dissertação é proposto um protocolo de comunicação sem-fios baseado numa tecnologia padrão de baixo consumo para um sistema de domótica habitacional. Este protocolo pretende possibilitar a comunicação entre dispositivos (sensores, actuadores e outros), permitindo a diferenciação da qualidade de serviço de acordo com o perfil de tráfego de cada fluxo de dados. Adicionalmente, o protocolo deve assegurar que as comunicações com requisitos de tempo-real exigentes não são afectadas por transmissões de outras tecnologias de contenção que operem sobre a mesma banda de frequências.

1.2 Contribuições

Esta dissertação apresenta a especificação e o desenvolvimento de um protocolo de comunicação sem-fios para domótica habitacional, capaz de integrar serviços de monitorização, localização e de registo de eventos. Neste âmbito, salientam-se as seguintes contribuições:

1. Definição de um protocolo de comunicações sem-fios baseado no WFTT (*Wireless Flexible Time-Triggered*), mas adaptado a um cenário de aplicação na área da domótica habitacional.
2. Proposta de um modelo de endereçamento adequado para aplicações de domótica, permitindo um nível elevado de flexibilidade na selecção dos dispositivos a que se destina uma mensagem.
3. Definição da arquitectura de software do protocolo B-Live Wireless e respectivo desenvolvimento das bibliotecas de suporte.
4. Definição de uma arquitectura de *hardware* com elevado potencial em termos da heterogeneidade de sensores e actuadores suportados, usando um módulo “inteligente” comum a todos os dispositivos.

1.3 Estrutura da dissertação

A dissertação encontra-se dividida em cinco capítulos. O primeiro capítulo apresenta a motivação do trabalho, os objectivos e as contribuições daí resultantes. No segundo capítulo é feito um levantamento do trabalho anterior na área das comunicações sem fios de curto alcance para aplicações de domótica. Neste capítulo, além de um levantamento das características principais de cada protocolo é realizada uma análise comparativa.

O protocolo B-Live Wireless é introduzido no terceiro capítulo, indicando as motivações de desenvolvimento e fornecendo informação pormenorizada relativa ao protocolo WFTT que o sustenta. Adicionalmente, são apresentados os serviços a suportar bem como os actores envolvidos no processo de instalação e utilização do sistema. A arquitectura, os elementos constituintes e o modelo de comunicação são também alvo de análise detalhada neste capítulo.

O quarto capítulo refere-se ao desenvolvimento do sistema. Neste sentido, definem-se as arquitecturas de *software* e *hardware*. Na primeira parte é descrita a implementação do protocolo tanto ao nível da organização de camadas como do código fonte correspondente. A segunda parte descreve os aspectos relacionados com a concepção do *hardware* de suporte ao protocolo B-Live Wireless.

Finalmente, o capítulo cinco encerra esta dissertação apresentando um resumo do trabalho realizado. Neste capítulo, discutem-se ainda as contribuições e os aspectos a considerar em trabalho futuro.

2. Trabalho Anterior

2.1 Introdução

O trabalho desenvolvido nesta dissertação consiste na criação de um novo protocolo de comunicação para domótica habitacional. Embora existam actualmente soluções comerciais para esta aplicação, cada uma apresenta vantagens e desvantagens específicas, sendo difícil encontrar uma solução que satisfaça globalmente os seus requisitos, em particular, se for considerada instalação em habitações concluídas.

Em seguida, apresenta-se um levantamento do estado da arte que tem como objectivo identificar as tecnologias mais relevantes e as suas características. Embora existam vários meios físicos de comunicação tais como redes cabladas (Ethernet, etc.), redes sem fios (Wi-Fi, etc.) ou redes *Power Line* (PLC), serão focadas apenas as soluções baseadas em tecnologias de comunicação sem fios de baixo consumo, dado ser este o enfoque da rede de comunicações locais do projecto LUL.

2.2 Sistemas de automação habitacional sem fios

Existem vários protocolos de automação habitacional que empregam tecnologias de comunicação sem-fios. No entanto, cada protocolo apresenta um conjunto de vantagens e desvantagens específico face aos seus concorrentes. De forma a melhor caracterizar as funcionalidades oferecidas pelos protocolos mais relevantes realizou-se uma análise aos cinco protocolos mais populares: *EnOcean*, *Insteon*, *ONE-NET*, *Z-Wave* e *ZigBee*. Em seguida, apresentam-se as suas características de forma resumida.

EnOcean

EnOcean [6] é uma tecnologia privada desenvolvida na Alemanha que permite a utilização de pequenos dispositivos sem bateria, através do processo de obtenção de energia designado por *energy harvesting*, onde a energia necessária para operar os dispositivos é derivada de fontes externas tais como dispositivos piezoeléctricos, células solares ou termopares.

Embora os dispositivos que operam com esta tecnologia não utilizem bateria, têm a capacidade de operar em modo P2P e cobrir distâncias até 300 metros com uma taxa de

transmissão máxima de 120kbps. Contudo, a tecnologia apenas permite a transmissão de pacotes de dados curtos. A tecnologia também não suporta segurança na transmissão de dados e a frequência em que opera apresenta limitações em termos da utilização livre do espectro na Europa¹ ao nível do *duty-cycle*. Na Tabela 1 são expostas as principais características deste protocolo.

Tabela 1: Características do protocolo EnOcean

Número de dispositivos	Tipo de topologia	Segurança	Tamanho dos pacotes	Taxa de transmissão	Frequência de operação	Alcance máximo
65.536	P2P	Não	14 bytes	120 kbps	868 MHz	300 m

Insteon

O protocolo *Insteon* [7] foi criado como um sucessor do anterior X10, que operava sobre uma rede *Power Line Communication* (PLC). Em teoria, os dispositivos a operar com este protocolo estão preparados para utilizarem ambos os suportes físicos: *wireless* e PLC. No entanto, a maior parte dos dispositivos comerciais apenas dispõe do primeiro.

O protocolo *Insteon* suporta um número elevado de dispositivos, operando numa topologia de rede do tipo *mesh*². Contudo, actualmente, apenas pode ser utilizado em redes de 110VAC (60HZ), significando que não podem ser instalados nos países Europeus. Além desta desvantagem, tanto a taxa de transferência como o tamanho dos pacotes é relativamente reduzido. A sincronização da rede é conseguida através da rede eléctrica, onde cada um dos dispositivos reenvia de novo para a rede as mensagens que recebe desta, durante um período de tempo fixo. Devido ao facto de poderem ocorrer colisões, existe a possibilidade de perda de pacotes, sendo esta potenciada em situações onde a existência de repetidores e filtros não se verifica. Na Tabela 2 são expostas as principais características deste protocolo.

Tabela 2: Características do protocolo Insteon

Número de dispositivos	Tipo de topologia	Segurança	Tamanho dos pacotes	Taxa de transmissão	Frequência de operação	Alcance máximo
16.777.216	<i>mesh</i>	Não	10 bytes	13.165 bps	915 MHz	50 m

¹ A frequência ISM de 868 MHz tem uma limitação de 1% de *duty cycle* na Europa, significando que os dispositivos apenas podem transmitir 1% do tempo. Esta limitação não é imposta nos Estados Unidos, mas é imposto um limite de potência inferior. [9]

² Uma topologia *mesh* é uma arquitectura em que vários dispositivos se ligam aos mais próximos, utilizando-os, caso necessário, para estabelecer comunicações com outros que se encontram fora de alcance. A mensagem é reencaminhada dispositivo após dispositivo desde a origem até ao destino.

ONE-NET

O protocolo *ONE-NET* [8] é um protocolo aberto que pode ser adaptado a um elevado número de dispositivos (ou sistemas), opera em diferentes topologias e dispõe de protecção permanente de dados baseada em encriptação. O protocolo caracteriza-se ainda por uma área de cobertura (alcance) elevada e largura de banda reduzida. A frequência de operação apresenta limitações em termos do *duty-cycle* máximo, tal como ocorre para o protocolo *EnOcean*. O protocolo *ONE-NET* permite controlar, no máximo, 4096 dispositivos, sem que estes possam ser agrupados relativamente às suas funcionalidades, tais como um grupo de dispositivos actuadores e um grupo de sensores. Na Tabela 3 são expostas as principais características deste protocolo.

Tabela 3: Características do protocolo ONE-NET

Número de dispositivos	Tipo de topologia	Segurança	Tamanho dos pacotes	Taxa de transmissão	Frequência de operação	Alcance máximo
4.096	estrela, P2P, <i>mesh</i>	Sim XTEA2-32	>14 bytes	38.4 kbps	868 MHz	500 m

Z-Wave

Z-Wave é um protocolo de código fechado que permite a criação de redes *mesh* reunindo até um máximo de 232 dispositivos. Este protocolo dispõe de uma largura de banda e área de cobertura reduzidas. O protocolo permite a atribuição de grupos aos diferentes dispositivos, para que, desta forma, seja possível juntar elementos com funcionalidades comuns, facilitando a comunicação e compreensão da rede. O protocolo *Z-Wave* possui suporte para sistemas PLC e permite controlo via dispositivos de controlo remotos comuns, encontrados em equipamento doméstico variado. De forma semelhante, tal como ocorre nos protocolos descritos, opera sobre uma gama de frequências com restrições de *duty-cycle* na Europa. A Tabela 4 mostra as principais características deste protocolo.

Tabela 4: Características do protocolo Z-Wave

Número de dispositivos	Tipo de topologia	Segurança	Tamanho dos pacotes	Taxa de transmissão	Frequência de operação	Alcance máximo
232	<i>mesh</i>	Sim	-	40 kbps	868 MHz	30 m

ZigBee

O *ZigBee* é uma tecnologia de comunicação de curto-alcance muito popular em aplicações de domótica. Suporta as topologias de rede em estrela e *mesh*, apresentando uma taxa de transferência de dados moderada. O protocolo pode operar na mesma banda de frequência do *Bluetooth* e *Wi-Fi*, a banda Industrial Científica e Médica (ISM) dos 2.4GHz, o que pode ser uma desvantagem devido às potenciais interferências destas tecnologias. O *ZigBee* possui suporte para comunicações seguras através de um mecanismo de encriptação (por hardware) ao nível do *transceiver*. Embora seja uma tecnologia bastante difundida e com suporte para diversos perfis de aplicação, carece de uma licença de utilização, o que pode constituir um impedimento para empresas de dimensão reduzida. Na Tabela 5 mostram-se as principais características deste protocolo.

Tabela 5: Características do protocolo ZigBee

Número de dispositivos	Tipo de topologia	Segurança	Tamanho dos pacotes	Taxa de transmissão	Frequência de operação	Alcance máximo
64535	estrela, <i>mesh</i>	Sim AES-128	-	250 kbps	2.4 GHz	30 m

Discussão

A análise aos protocolos descritos permite estabelecer um termo de comparação, mostrando as diferentes vantagens e desvantagens entre tecnologias. No que respeita ao número de dispositivos suportados pela rede, o protocolo *Insteon* apresenta vantagem clara pois é capaz de suportar uma rede com 16.777.216 elementos. A possibilidade de permitir que o instalador possa criar grupos, oferecendo desta forma uma estrutura mais simples à rede, é também uma vantagem.

A topologia da rede é um aspecto que depende substancialmente do modelo de comunicações adoptado. As topologias mais comuns a suportar são a P2P e a *mesh*, estas duas opções são suportadas no protocolo *ONE-NET* e no protocolo *ZigBee*, os restantes apenas admitem uma delas.

A segurança é um aspecto crítico em redes de comunicação sem-fios. Neste sentido, não deve permitir um controlo mal-intencionado, por parte terceiros, dos dispositivos existentes na habitação. Os protocolos *ONE-NET*, *Z-Wave* e *ZigBee* oferecem suporte de encriptação que impede tanto a divulgação de informação como a injeção de informação maliciosa na rede.

Quando os dispositivos trocam informação entre si, os dados são encapsulados em pacotes e transmitidos no ar. O tamanho dos pacotes suportados pelas diferentes tecnologias varia bastante. Em aplicações de automação habitacional não existem requisitos exigentes de *throughput*. Por esta razão, pode-se considerar que pacotes com tamanhos entre 15 e 25 bytes são suficientes para cumprir com as necessidades da aplicação. Contudo, de forma a suportar um leque mais abrangente de cenários de aplicação, é prudente utilizar uma tecnologia com

capacidade para integrar *payloads* maiores. Apenas as tecnologias *ONE-NET*, o *Z-Wave* e *ZigBee* permitem um tamanho de pacote superior a 15 bytes.

Outro aspecto relevante é a taxa de transmissão. No entanto, verifica-se uma forte correlação (inversa) entre esta característica e o consumo de energia. Um sistema de domótica habitacional que opera com dispositivos alimentados a baterias tem a autonomia como um requisito fundamental. No entanto uma taxa de transmissão mais elevada garante, para o mesmo número de mensagens, uma ocupação de canal muito mais reduzida. Desta forma garante-se também um atraso bastante inferior no que toca à latência entre um emissor e o respectivo receptor. Dos protocolos analisados apenas o *EnOcean* e o *ZigBee* têm capacidade para suportar elevadas taxas de transferência.

Em relação à frequência de operação, todos os protocolos expostos apresentam uma vantagem e uma desvantagem. Inicialmente surge a necessidade de excluir o protocolo *Insteon*, pois opera na frequência de 915MHz, não permitida na Europa. Os restantes operam na banda livre ISM a uma frequência de 868MHz ou a 2.4GHz. A frequência de 868MHz apresenta uma limitação importante, que já foi descrita juntamente com a exposição do protocolo *EnOcean*, isto é, a impossibilidade de transmissão de pacotes com *duty-cycle* superior a 1% [9]. No entanto, actualmente, é uma banda menos populada que a banda ISM dos 2.4GHz. Esta última tem a particularidade de ser suportada globalmente. No entanto, permite várias tecnologias de comunicação, entre elas, Bluetooth e Wi-Fi. Estas tecnologias podem ser a fonte de interferência e ruído indesejado, podendo mesmo comprometer as comunicações das tecnologias de baixo consumo com que partilham o meio.

No que diz respeito ao alcance máximo dos dispositivos, dado a aplicação alvo, torna-se desnecessária uma tecnologia com um alcance de comunicação elevado pois isso implicaria um maior consumo de energia.

Finalmente, a questão do licenciamento dos protocolos. Na construção de uma solução comercial o pagamento de uma licença tem como impacto um aumento nos custos de cada dispositivo, o que não ocorre num sistema de código aberto. Este tipo de solução, além de resultar num menor custo, permite uma maior liberdade ao programador para adaptar a solução ao fim pretendido.

Pode concluir-se que nenhum dos protocolos enumerados reúne todas as características necessárias para se adaptar à aplicação alvo. É necessário suporte para um elevado número de dispositivos e a possibilidade de os agrupar. Devem ser suportadas diversas topologias, tais como P2P, estrela e *mesh*. As comunicações entre dispositivos devem ser encriptadas para garantir segurança, impedindo o acesso à informação por parte de estranhos. Os pacotes de dados devem ter um tamanho superior a 15bytes de forma a possibilitar o envio de informação "rica" e possibilitar a futura extensão de funcionalidades. A tecnologia seleccionada dever ser capaz de garantir comunicações em tempo-real para a aplicação alvo e, dadas as limitações da banda de 868MHz, operar na banda de ISM dos 2.4GHz. O suporte de comunicações de tempo-real deve ocorrer ainda que exista interferência de outras tecnologias de comunicação.

2.3 B-Live

O protocolo desenvolvido nesta dissertação tem as suas raízes em outro protocolo denominado B-Live. O B-Live é uma solução de domótica, criada pela Micro I/O em 2006, destinada à adaptação de casas convencionais, permitindo que pessoas com graves limitações funcionais, por exemplo tetraplégicos, possam controlar com autonomia, ainda que limitada, algumas funcionalidades da sua habitação (por exemplo, acesso, iluminação e operação de dispositivos). O projecto, tal como foi referido, foi vencedor do Prémio Eng. Jaime Filipe em 2007, encontrando-se implementado em vários locais, dos quais se destaca o Centro de Medicina de Reabilitação do Centro e a Escola Superior de Saúde de Coimbra.

O protocolo de comunicação do projecto B-Live opera usando o barramento *Controller Area Network* (CAN), incluindo um modo de compatibilidade RS232. As tramas são essencialmente compostas por: um campo referente ao endereço de origem, definido como "IDSrc" (11 bits); um campo que define o comprimento do campo de dados da trama, definido como "L" (8 bits); finalmente oito campos de dados, desde "D0" até "D7", contendo 8 bits cada. A Figura 1 mostra a constituição de uma trama do protocolo B-Live, utilizada no barramento CAN.



Figura 1: Trama do protocolo B-Live implementado em CAN

A trama pode ser adaptada ao protocolo série RS232, bastando para isso adicionar o carácter "%" no início da trama, e o campo de *checksum* "CRC", juntamente com o carácter "#" no final da trama. A Figura 2 demonstra este cenário.



Figura 2: Trama do protocolo B-Live implementado em RS232

As mensagens são transmitidas no barramento em *broadcast*, ou seja, todos os dispositivos receberão as mensagens, no entanto, apenas os seleccionados através do endereço irão executar as acções solicitadas. Este modelo de comunicações é conhecido como produtor-consumidor.

Relativamente aos campos de dados, o primeiro campo (*D0*) define o tipo de comando a executar, é definido por um carácter *ASCII* (por exemplo: C= configuração; D=porta; A=alteração de estado, etc.). O segundo campo de dados (*D1*) representa o tipo de módulo (carácter *ASCII*, por exemplo: L=lâmpada; S=interruptor; O=tomada, etc.). Os campos restantes (*D0* a *D7*) definem o identificador do dispositivo a controlar, exceptuando-se a situação onde *D0* indica uma alteração de estado.

Dado ser suportado sobre um meio físico cablado, o protocolo CAN apresenta características de fiabilidade e pontualidade muito favoráveis. Um exemplo disso é a sua aplicação em áreas exigentes tais como a indústria automóvel e a aviónica. No entanto, na área da domótica, em particular em cenários de adaptação de casas convencionais, esta tecnologia mostra-se menos apelativa devido à instalação de novas condutas para a cablagem que tornam a instalação mais cara e menos apelativa. O protocolo B-Live suportado sobre este meio físico apresenta ainda os seguintes inconvenientes:

- Os endereços de origem e de destino não se encontram definidos nas tramas, representando uma limitação ao nível da autenticação de mensagens no sistema;
- O endereçamento dos dispositivos está condicionado à utilização de 8 áreas, contendo 256 dispositivos cada uma (resulta dos 11 bits utilizados). Dado que muitas habitações dispõem de um número superior de divisões, este é um factor que impede a aplicabilidade generalizada;
- A utilização de caracteres *ASCII* como códigos é limitadora ao nível do número de operações que permite representar;
- Não existem campos livres que possam ser utilizados para estender o protocolo, mantendo a compatibilidade com as versões anteriores.

Estes factores, em conjunto com as limitações da utilização do barramento CAN e a necessidade de desenvolver uma solução de domótica integrada com suporte para vários tipos de serviços motivaram o desenvolvimento de uma versão *wireless* deste protocolo, como se apresenta em seguida.

3. Protocolo B-Live Wireless

A secção anterior apresentou várias soluções tecnológicas para domótica habitacional. Através da sua comparação, foi possível verificar as vantagens e desvantagens da sua utilização. Um aspecto comum a todas elas é vulnerabilidade a interferência. Dado que se pretende um protocolo de comunicação sem-fios capaz de suportar fluxos de dados com QoS garantida em ambientes abertos, estas soluções não são directamente aplicáveis. Em seguida, concluiu-se que a melhor solução seria o estudo, desenvolvimento e implementação de um novo protocolo de comunicação, com recurso à utilização de algumas características que as tecnologias anteriores já dispunham e com a oferta de um número mais elevado de serviços aos utilizadores, descritos nos capítulos seguintes.

3.1 Serviços oferecidos

Dependendo do actor que interage com o sistema B-Live Wireless, vários serviços podem ser oferecidos, englobando a execução de acções, monitorização de eventos ou obtenção de informação. Podem-se considerar três actores que interagem com o sistema:

- I. Instalador
- II. Utilizador
- III. Cuidador

- I. O instalador é o elemento responsável pela implementação e configuração do sistema na habitação. Neste âmbito, o sistema deve oferecer ao instalador serviços simples e intuitivos que permitam a:
 - identificação de dispositivos;
 - alteração de endereçamento ou agrupamento de dispositivos;
 - configuração e programação de opções gerais e locais.
- II. O utilizador interno corresponde à pessoa (ou o grupo de pessoas) que utilizam efectivamente o sistema. Fazem parte deste grupo os habitantes da área monitorizada. A estes, deverão ser oferecidos os seguintes serviços:

- actuação sobre dispositivos da habitação (por exemplo, efectuar a operação de ligar/desligar uma lâmpada através de um interruptor ou outro dispositivo de entrada);
 - monitorização ambiente (por exemplo, luminosidade ou humidade);
 - monitorização de segurança (como alarmes).
- III. O cuidador é a pessoa ou o grupo de pessoas com acesso ao sistema para monitorizar o estado da habitação e dos seus utilizador, tais como familiares, centros hospitalares ou unidades de intervenção de emergência. Neste âmbito, devem ser suportados os seguintes serviços:
- localização dos utilizadores na habitação;
 - monitorização do estado de saúde e condição física em tempo real;
 - monitorização do estado da habitação (estado das janelas, da iluminação, ou dos alarmes);
 - geração automática de alarmes por ocorrência de situações anormais.

3.2 Requisitos

O desenvolvimento do sistema B-Live Wireless comporta requisitos a vários níveis, em particular, ao nível das comunicações sem-fios que o suportam. Neste sentido, o protocolo de comunicações empregue deve ser capaz de suportar três tipos de tráfego, a saber:

- *Hard real-time*
- *Soft real-time*
- *Best-Effort*

No primeiro nível, encontram-se todas as comunicações que necessitam de muito baixa latência nas transmissões. São exemplos os dispositivos que monitorizam os sinais vitais de um paciente ou a rede de sensores da habitação que pode gerar alarmes (detecção de fumos, gás, inundação, entre outros). Estes alarmes podem ser dirigidos a familiares, a um centro hospitalar ou a unidades de intervenção de emergência (bombeiros, polícia, entre outros).

No segundo nível, encontram-se comunicações menos *time-critical*. Fazem parte deste grupo as comunicações provenientes de monitorização e actuação de dispositivos tais como o controlo de um candeeiro através de um interruptor ou a abertura de uma porta. Têm como objectivo o pedido de uma acção e a sua respectiva execução. Embora sejam de menor prioridade, os atrasos existentes devem ser imperceptíveis para o utilizador.

Finalmente, no nível inferior operam as comunicações menos prioritárias designadas por *Best-Effort*. Exemplos destas comunicações são a monitorização de eventos do utilizador (os serviços de localização, a detecção de presença em sofás ou outros elementos da habitação). Têm como objectivo apenas o registo de eventos numa base de dados e a subsequente disponibilização em serviços pouco críticos.

O suporte destes requisitos tem nas comunicações *hard real-time* o seu maior desafio. Tal ocorre devido à vulnerabilidade das tecnologias de baixo consumo a transmissões de outras tecnologias de maior potência que operem na vizinhança e sobre a mesma banda de frequências. Assim, de forma a contornar este problema, desenvolveu-se um protocolo inspirado no protocolo WFTT. Em seguida, faz-se uma breve descrição do seu funcionamento.

3.3 Protocolo WFTT

A tecnologia de comunicação IEEE 802.15.4 opera na banda dos 2.4GHz, tipicamente populada por dispositivos *ZigBee*, *Wi-Fi* e *Bluetooth*. Uma solução utilizada para contornar o problema da interferência de outras tecnologias, quando um dispositivo deseja efectuar as suas comunicações, é designada por *bandjacking* [10]. Esta solução é implementada no protocolo *Wireless Flexible Time-Triggered* (WFTT) [11].

O *bandjacking* é uma técnica que permite forçar o acesso a um canal de comunicação, mesmo que outras tecnologias se encontrem a comunicar nesse mesmo canal. O funcionamento desta técnica é descrito de seguida.

Uma transmissão em tempo-real é precedida por um *black-burst* [12] com uma duração temporal fixa o qual é transmitido por um coordenador (*Master*) da rede, fazendo com que todas as restantes comunicações sejam interrompidas. Caso esta duração seja suficientemente longa, os restantes dispositivos vêem o canal como ocupado e entram em modo de espera, tal como demonstrado na Figura 3.

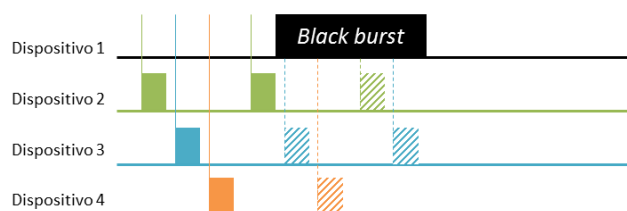


Figura 3: Interrupção de transmissões devido a um *black-burst*

Após o final do *black-burst*, o coordenador é responsável pela transmissão de um *trigger packet*, onde constam todas as informações referentes a ordens e prioridades dos dispositivos que necessitam de efectuar as suas comunicações. O pacote pode ser visto na Figura 4 e apresenta-se a vermelho com a letra "P". Todos os dispositivos da rede recebem o pacote, após este ter sido devidamente processado, e, com base nele, as prioridades são atribuídas.

Imediatamente após, uma janela PW³ é aberta e reservada para comunicações de elevada prioridade. O dispositivo com maior prioridade, identificado como "A", executa a sua

³ PW é um acrónimo para *Protected Window*, é uma janela de tempo reservada exclusivamente para comunicações prioritárias multiplexadas no tempo (TDMA)

comunicação, precedendo o dispositivo “B”. Este formato de transmissão é denominado por TDMA⁴ pois respeita intervalos de tempo estritamente definidos.

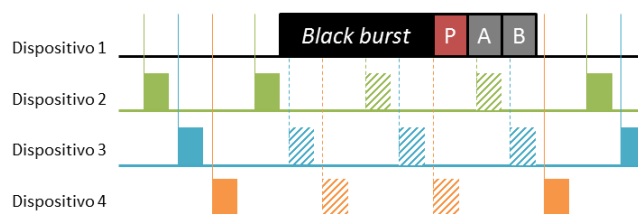


Figura 4: Transmissão de dados segundo o protocolo WFTT

Finalmente, após todas as comunicações prioritárias terem sido efectuadas, a janela PW é fechada, dando início à segunda fase do processo. Nesta fase, é aberta uma nova janela para comunicação, designada por CW⁵. Os dispositivos não prioritários iniciam, então, as suas comunicações segundo um mecanismo de pré-escuta designado por CSMA/CA⁶.

Ao fim de um determinado período de tempo a janela CW é fechada, dando-se por terminado um ciclo elementar completo (EC⁷) de comunicação. Quando for pretendido, o coordenador dá início a um novo ciclo elementar começando com um novo *black-burst*.

A Figura 5 ilustra todas as partes constituintes de uma trama de acordo com o protocolo WFTT. Podem observar-se a janela de *black-burst*, o *trigger packet* (P), a *protected window* (PW) e a *contention window* (CW). Todos estes elementos formam uma *superframe* que é repetida a cada *elementary cycle* (EC).

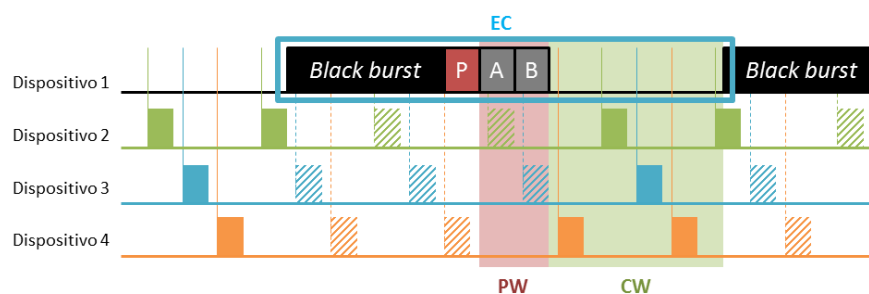


Figura 5: As diferentes partes constituintes de uma trama WFTT

Desta forma é possível efectuar comunicações de dados em tempo-real, mesmo que o sistema esteja perante um ambiente onde o ruído seja abundante.

⁴ TDMA é um acrónimo para *Time Division Multiple Access* e, é um método em que um canal é dividido em intervalos de tempo distintos, permitindo assim que vários dispositivos possam comunicar sem interferências.

⁵ CW é um acrónimo para *Contention Window*, trata-se de uma janela de tempo reservada para comunicações livres sem intervalos temporais pré-definidos. Os dispositivos aplicam o método CSMA para comunicar nesta janela.

⁶ CSMA/CA é um acrónimo para *Carrier Sense Multiple Access with Collision Avoidance*. É um algoritmo, o qual verifica se o canal se encontra livre antes de proceder ao início da comunicação, Caso não se verifique essa situação, aguarda um tempo aleatório.

⁷ EC é um acrónimo para *Elementary Cycle*. Este ciclo elementar é composto pelo *black-burst*, pela *protected window* (PW), pela *contention window* (CW) e por uma janela inactiva – *inactive window* (IW), sendo esta última opcional.

3.4 Solução

De acordo com as tecnologias estudadas no capítulo 2.2 , foi possível verificar que nenhuma delas se adequa a um ambiente com os requisitos identificados. Apesar de terem bastantes pontos favoráveis, não são suficientemente robustas para serem aplicadas neste projecto. A solução encontrada baseia-se no protocolo WFTT descrito no capítulo 3.2 .

Como solução para este problema, a aplicação do protocolo WFTT descrito no capítulo 3.2 , revela-se uma escolha bastante eficaz, pois consegue-se obter uma garantia relativamente à execução de comunicações de tempo-real. Desta forma, unindo todos os pontos favoráveis em cada uma das tecnologias descritas atrás, assim como o protocolo WFTT, optou-se por desenvolver o novo protocolo de comunicação B-Live Wireless.

3.4.1 Arquitectura

O sistema B-Live Wireless não foca exclusivamente o controlo habitacional, mas também a monitorização de utilizadores e a detecção de eventos. Em consequência, tornou-se necessário agrupar os dispositivos constituintes do sistema em três grandes categorias: monitorização, controlo e detecção de eventos.

Na categoria de monitorização são englobados todos os dispositivos destinados à monitorização do utilizador e da habitação:

- medição de sinais vitais;
- localização do utilizador dentro da habitação;
- consumo energético (como electricidade, água e gás);
- medição de temperatura.

Na categoria de controlo são incluídos dispositivos actuadores que têm a capacidade de executar tarefas específicas. Alguns exemplos dessas tarefas são:

- abertura ou fecho de portas ou janelas;
- abertura ou fecho de estores;
- controlo de iluminação;
- controlo de tomadas;
- dispensadores automáticos;
- sistema de controlo de gás.

Finalmente, na categoria de detecção de eventos, encontram-se reunidos os sensores que têm como função detectar o estado, ou a alteração deste, em elementos existentes na habitação. Fazem parte desta categoria elementos para:

- a detecção de abertura ou fecho da porta;
- Interruptores de iluminação sensíveis ao toque;
- a detecção de presença humana (em sofás ou camas);
- a detecção de fumos ou fugas de gás;
- a detecção de fluxo de água.

Os sensores e os actuadores são constituídos por um pequeno módulo sem fios de baixa potência que, permite a transferência de informação entre os dispositivos e um servidor. As comunicações da rede são controladas por esse servidor através de *Gateways*⁸, os quais são igualmente responsáveis por estabelecer a ponte das comunicações da rede sem fios de sensores e actuadores: *Wireless Sensor/Actuator Network* (WSAN).

A arquitectura do sistema é baseada numa rede de duas camadas, tal como pode ser observado na Figura 6. A camada inferior é implementada através de uma rede sem fios pessoal (WPAN⁹) de acordo com o *standard* IEEE 802.15.4, enquanto a camada superior é baseada numa rede cablada (LAN¹⁰) ou sem fios (WLAN¹¹), tipicamente Wi-Fi, definidas pelos *standards* IEEE 802.3 ou o IEEE 802.11, respectivamente.

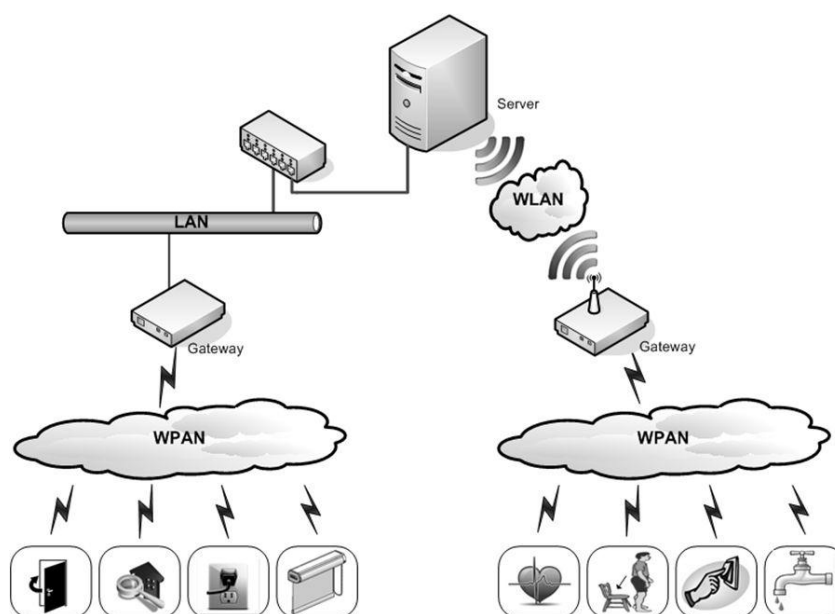


Figura 6: Arquitectura da rede B-Live Wireless

As duas camadas da rede estão ligadas entre si através dos *Gateways*, permitindo a troca de informações entre o servidor e os dispositivos da WSAN (WPAN).

As informações adquiridas pela WSAN são monitorizadas pelos *Gateways* e transmitidas ao servidor, onde podem ser processadas no momento ou guardadas numa base de dados, para uma posterior análise.

⁸ Os *Gateways* são dispositivos destinados a interligar duas redes com características diferentes, tais como diferentes suportes físicos ou protocolos.

⁹ WPAN é um acrónimo para *Wireless Personal Area Network*, uma rede de dispositivos de curto alcance, semelhante ao *Bluetooth*.

¹⁰ LAN é um acrónimo para *Local Area Network*, uma rede local cablada, geralmente a operar sobre o protocolo IEEE 802.3. As ligações comuns de 10Mbit, 100Mbit e 1Gbit presentes nos computadores pessoais são exemplos de LAN's e surgem de uma forma comum com o nome Ethernet.

¹¹ WLAN é um acrónimo para *Wireless LAN*. As redes tipicamente designadas por Wi-Fi e utilizadas nos computadores pessoais são um exemplo de WLAN's.

3.4.2 Operação

A operação do sistema B-Live Wireless pode ser resumida através de um simples exemplo. Quando um utilizador da habitação se senta numa cadeira (equipada com um sensor adequado), esta detecta uma alteração de estado e gera um evento que resulta na transmissão de um pacote IEEE 802.15.4 na WSN. Ao receber este pacote, o *Gateway* reencaminha os dados recebidos para o servidor utilizando uma das duas alternativas descritas anteriormente: IEEE 802.3 (LAN) ou IEEE 802.11 (WLAN). Os dados são recebidos no servidor e guardados numa base de dados, permitindo disponibilizar a informação para consulta posterior.

Caso seja necessário, o servidor pode ordenar a operação de um, ou vários actuadores específicos, levando à execução de uma determinada acção, por exemplo o controlo de uma lâmpada ou a abertura de uma porta na habitação. Os dados recolhidos da WSN e armazenados no servidor são essenciais para aplicações como a localização de um utilizador dentro da habitação ou a monitorização dos seus sinais vitais. Estes dados podem ser utilizados para estudar o movimento e o comportamento dos utilizadores permitindo, também, detectar antecipadamente ocorrências anómalas na habitação ou ao nível da saúde dos utilizadores.



Figura 7: Processo de registo e análise de eventos na rede

A informação registada no servidor é também disponibilizada a aplicações ou utilizadores externos autorizados, através de um *website* ou de um serviço *web*.

3.5 Elementos do sistema

A rede local de sensores e actuadores (WSN) contém vários dispositivos interligados entre si, através de uma ligação sem fios baseada no *standard* IEEE 802.15.4. Estes dispositivos podem ser agrupados nas seguintes categorias:

- Actuadores
- Sensores
- Controladores
- *Gateway*

Actuadores

Os actuadores são dispositivos que permitem controlar elementos específicos. Alguns exemplos são: um sistema de porta automática, onde o motor é controlado por um actuador; um controlador de iluminação, onde o actuador liga ou desliga a lâmpada; um controlador de uma torneira, no qual o actuador é responsável por controlar uma válvula; um controlador de estores, no qual, novamente o motor é controlado por um actuador; um microondas ou um forno eléctrico, ambos controlados por tomadas que incorporam actuadores; uma ventoinha ou mesmo um sistema de aquecimento.

Os actuadores são, regra geral, alimentados pela rede eléctrica, não necessitando, obrigatoriamente, de bateria. No entanto esta pode ser adicionada em casos que assim o exijam.

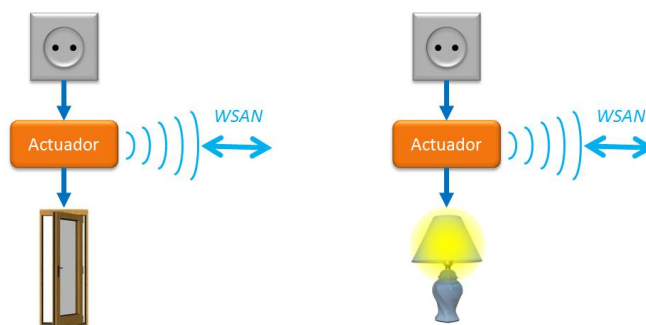


Figura 8: Os actuadores controlam outros dispositivos da habitação

Sensores

Os sensores são utilizados de forma a obter informações acerca do ambiente controlado, da interacção do utilizador com elementos da habitação ou do próprio utilizador. Estes dispositivos são divididos nas seguintes subcategorias:

- Sensores de evento: são dispositivos que geram um evento sempre que uma condição específica é cumprida, tais como interruptores, controlos remotos, sensores de portas e janelas, sensores de estado, sensores de pressão, sensores de queda ou detecção de interacção humana com um objecto.
- Sensores de monitorização: são dispositivos que monitorizam de forma contínua o ambiente, tais como sensores de temperatura, luz, gás ou água, ou que monitorizam utilizadores através de sensores de localização *tag* ou de medidores de sinais vitais.



Figura 9: Os sensores obtêm informações sobre o ambiente controlado

Controladores

Os controladores são dispositivos que se destinam à interligação de toda a rede de sensores e actuadores com o *gateway*. De forma a reduzir custos e espaço ocupado, este tipo de dispositivos pode ser implementado juntamente com actuadores, como lâmpadas de tecto ou tomadas de parede.

Os controladores desempenham um papel crucial na tarefa de localização do utilizador dentro da habitação, pois são os responsáveis pelo fornecimento das informações necessárias ao servidor, como será descrito adiante.

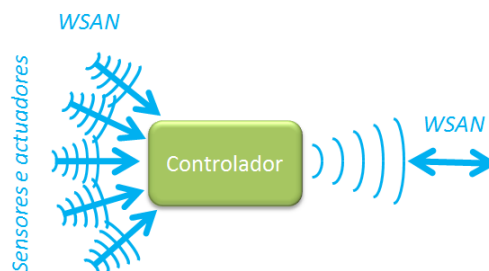


Figura 10: O controlador interliga os sensores e actuadores ao *gateway*

Gateway

O *gateway* é o dispositivo que efectua a ponte entre a rede WSAN a operar sobre o *standard* IEEE 802.15.4 e o servidor, localizado dentro ou fora da habitação, através de uma ligação *Ethernet* LAN ou Wi-Fi.

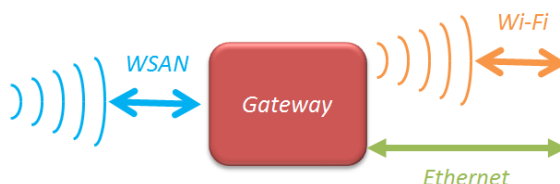


Figura 11: O *gateway* cria uma ponte entre a WSAN e o servidor

3.6 Modelo de comunicação

A Figura 12 ilustra a estrutura e as ligações entre os dispositivos descritos no subcapítulo anterior. Como pode ser observado, quando um sensor necessita de comunicar com o servidor, pode utilizar elementos intermédios para encaminhar a mensagem até ao destino. Neste caso, a mensagem é enviada para o controlador ao qual o dispositivo se encontra associado, este reencaminha-a para o *gateway* que, por sua vez, a transmite para o servidor. O mesmo processo é repetido em ordem inversa no caso de se tratar de um actuador.

Os sensores e actuadores podem também tomar decisões e executar tarefas de forma independente, ou seja, sem qualquer autorização do servidor. O controlador é utilizado para reencaminhar os dados caso seja necessário. No entanto, o servidor deve ser sempre informado das alterações efectuadas. Para além dos controladores, os únicos dispositivos autorizados a comunicar directamente com o *gateway* são os sensores de sinais vitais, devido à sua elevada prioridade e operação em tempo-real.

Os dispositivos *tag* são os responsáveis pela localização dos utilizadores dentro da habitação e, por isso, devem ser transportados pelos utilizadores. As informações transmitidas pelos dispositivos *tag* são recebidas por todos os controladores ao seu alcance.

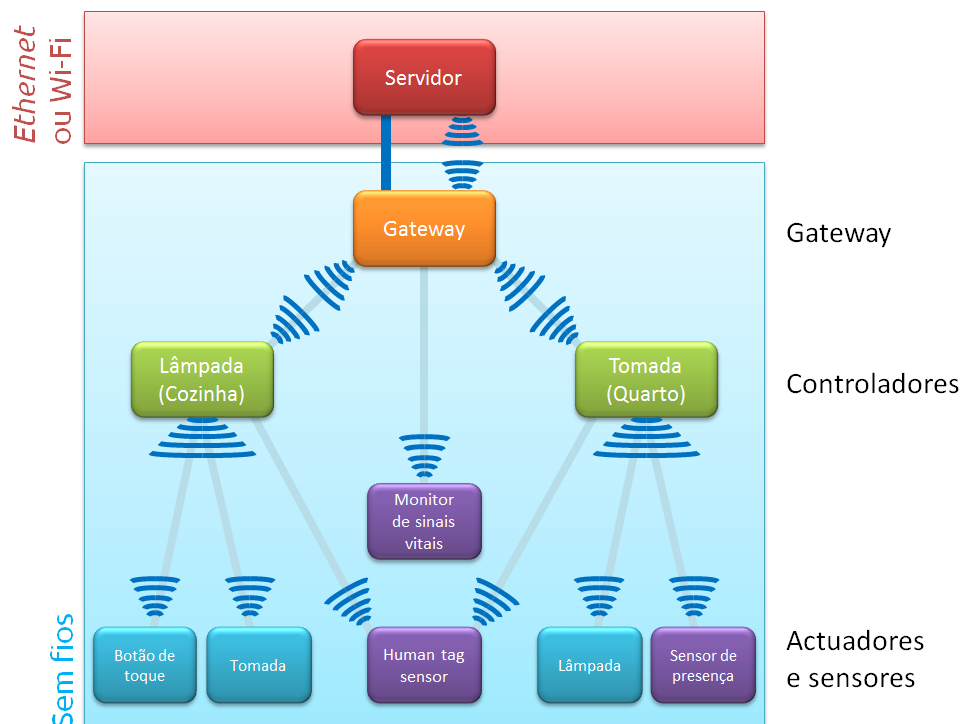


Figura 12: Estrutura de ligações entre as diferentes categorias de dispositivos

As comunicações sem-fios efectuadas na rede WPAN utilizam o modelo produtor-consumidor. Os produtores efectuam transmissões de pacotes para o endereço de *broadcast*, sendo os mesmos recebidos por todos os dispositivos que se encontrem à escuta na vizinhança.

Compete ao(s) dispositivo(s) receptor(es) identificar a fonte do pacote (produtor) e a informação correspondente e verificar se é um consumidor dessa mesma informação. Se for, deve responder ao pacote recebido. Caso contrário, deve ignorá-lo. A associação entre produtores e consumidores é realizada na fase de instalação do sistema B-Live Wireless.

Alguns exemplos em concreto serão expostos nos próximos capítulos, nos quais a ordem de comunicações será identificada de forma mais pormenorizada.

3.6.1 Características

O protocolo de comunicação assenta essencialmente sobre o *standard* IEEE 802.15.4, tendo sido desenvolvido principalmente para comunicações *peer-to-peer*.

No Anexo 1 é exposta e descrita a estrutura elementar de um pacote MAC baseado na norma IEEE 802.15.4. De acordo com esse pacote, o protocolo B-Live Wireless insere-se inteiramente no campo de dados *payload*, necessitando de operar em modo *broadcast*. Isto significa que todos os dispositivos recebem os dados, embora apenas os seleccionados possam processá-los, tal como será descrito mais a frente.

O protocolo B-Live Wireless dispõe do seu próprio sistema de endereçamento. Desta forma os campos de endereçamento existentes na norma IEEE 802.15.4 não são utilizados. Outra grande vantagem desta característica é a facilidade de aplicação do protocolo sobre outro suporte, como pode ser observado na Figura 13, fazendo com que o B-Live Wireless seja visto como um bloco único e independente da tecnologia que utiliza para a sua difusão. Além disso, a existência de encriptação permite manter seguros não apenas os dados, mas também os endereços dos dispositivos.

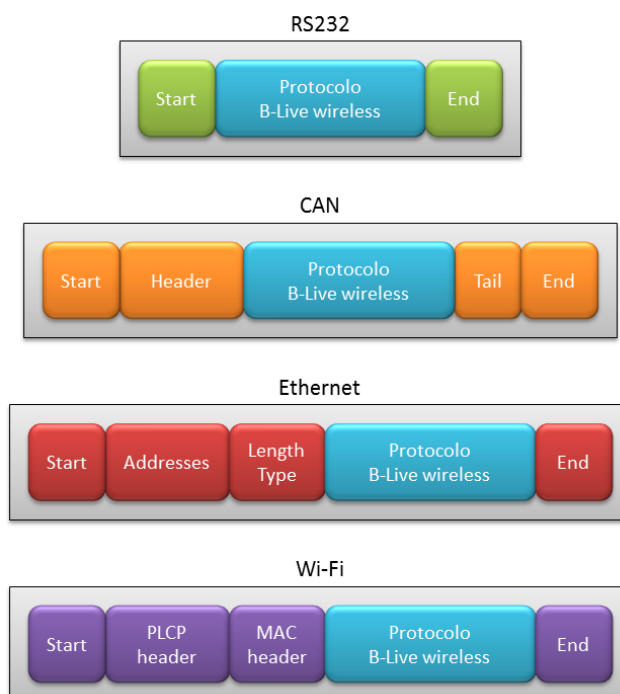


Figura 13: Facilidade de adaptação do protocolo B-Live Wireless a outros protocolos

Em comparação com as tecnologias estudadas no capítulo 2.2 , este novo protocolo tenta englobar o que de melhor existe nas tecnologias actualmente existentes no mercado. De entre esses pontos, podem distinguir-se:

- endereçamento de 32 bits, permitindo cerca de 4.300.000.000 dispositivos no mesmo sistema;
- o endereçamento de cada dispositivo pode ser dividido em campos: sector, área, grupo e identificação, permitindo assim uma localização precisa na habitação e uma definição exacta do tipo de dispositivo em questão;
- reencaminhamento de mensagens através de dispositivos definidos como controladores;
- possibilidade de controlo de vários dispositivos numa única transmissão;
- suporta pacotes de configuração, estado do dispositivo, bateria e depuração dos dispositivos;
- suporta dispositivos de tempo-real graças à tecnologia WFTT, tais como monitorização de queda ou de sinais vitais;
- possibilita a localização de vários utilizadores dentro da habitação;
- proporciona a utilização de um servidor para registar todas as actividades numa base de dados;
- raio de alcance de cerca de 100 metros em exterior, 30 no interior;
- muito baixo consumo energético;
- velocidade de transmissão de 250kbps;
- segurança de dados através de encriptação AES-128, onde todos os dados do protocolo são encriptados, incluindo os endereços dos dispositivos;
- frequência de operação de 2.4GHz, possibilitando que o sistema trabalhe em todo o mundo.

3.6.2 Estrutura elementar de uma trama

As tramas de comunicação do protocolo B-Live *wireless* contêm campos de tamanho estático e dinâmico. Na Figura 14 é apresentado o esboço de uma trama genérica, onde cada bloco representa 1 byte.

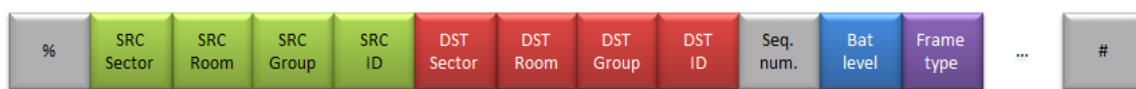


Figura 14: Estrutura de uma trama genérica

Os campos representados a cor cinzenta referem-se a campos de controlo do pacote: O carácter de início e de fim de trama são sempre constantes e são utilizados para verificação da autenticidade dos dados recebidos. O campo de *Sequence Number* é um campo de controlo da trama que a identifica como sendo única.

A cor verde encontram-se os campos referentes ao endereço do dispositivo que originou a mensagem e a cor vermelha, os referentes ao de destino. Estes campos encontram-se expostos, de uma forma mais pormenorizada, no subcapítulo 3.6.3 .

O nível de carga da bateria é transmitido em todas as tramas, sendo este designado por *Bat level*. Pode tomar valores entre 0x00_{hex} e 0xFF_{hex} segundo a Tabela 6. A transmissão desta informação é de grande importância dado que nos encontramos a trabalhar com dispositivos alimentados por bateria.

Tabela 6: Descrição dos valores associados ao campo *Bat level*

Bat level – Nível de bateria do dispositivo	
Valor (hex)	Descrição
0x00	Dispositivo a operar sem bateria
0x01 a 0xFE	Nível de carga da bateria
0xFF	Dispositivo com problemas na bateria

O campo *Frame type* identifica o tipo de pacote que foi recebido. Os campos seguintes até ao carácter de fim de trama são considerados como dinâmicos e dependem directamente do tipo de trama em questão. Esta parte é explicada, com maior pormenor no subcapítulo 3.6.4 .

3.6.3 Endereçamento

Endereçamento de origem

O endereçamento de origem é composto por 4 *bytes* e é dividido em quatro campos: *Sector*, *Room*, *Group* e *ID*. Desta forma, possibilita-se a existência de 4.294.967.296 endereços possíveis.



Figura 15: Campos de endereçamento de origem

SRC Sector

O campo de *Sector* contém o valor correspondente ao sector onde o dispositivo se encontra. Podem ser definidos 252 sectores de acordo com a Tabela 7. No caso de um bloco habitacional ou de um dormitório este campo pode definir o número da habitação ou do andar, para uma compreensão mais fácil.

Tabela 7: Descrição dos valores associados ao campo *SRC Sector*

SRC Sector – Sector do dispositivo de origem	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0xFC	Sector do dispositivo
0xFD	Dispositivo móvel
0xFE	Dispositivo de controlo geral (<i>Global</i>)
0xFF	Reservado

No caso de o dispositivo nunca ter sido configurado, este encontra-se por defeito com o valor 0x00_{hex}. Se se estiver perante um dispositivo móvel, como uma *tag* de localização, este campo terá um valor fixo definido entre 0x01_{hex} e 0xFC_{hex} na situação em que apenas opera num dado sector, ou terá o valor 0xFD_{hex}, quando é suposto operar independentemente do sector no qual se encontra. No caso de o valor ser 0xFE_{hex}, o dispositivo é considerado como sendo de controlo geral, abrangendo elementos como servidores, programadores ou *gateways*.

SRC Room

O campo de *Room* define a divisão da habitação em que o dispositivo de origem se encontra. Podem ser definidas 239 divisões de acordo com a Tabela 8. A tabela de divisões completa encontra-se no Anexo 2.

Tabela 8: Descrição dos valores associados ao campo *SRC Room*

SRC Room – Divisão do dispositivo de origem	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0x07	Reservado
0x08 a 0xF7	Divisão do dispositivo (consultar o Anexo 2)
0xF8 a 0xFC	Reservado
0xFD	Dispositivo móvel
0xFE	Dispositivo de controlo de sector (<i>Sector</i>)
0xFF	Reservado

No caso de o dispositivo nunca ter sido configurado, este campo encontra-se, por defeito, com o valor 0x00_{hex}. Para definir a divisão deve ser utilizado um valor da tabela do Anexo 2. No caso de um dispositivo móvel, o valor 0xFD_{hex} deve ser utilizado. Na eventualidade de estar perante um dispositivo de controlo de um sector específico, o campo *Sector* deve estar definido com o valor correspondente e aqui deverá ser definido o valor de 0xFE_{hex}. Esta situação é explicada mais adiante neste subcapítulo.

SRC Group

O campo de *Group* define a que grupo o dispositivo pertence. Os grupos são definidos de acordo com as funcionalidades que os diferentes dispositivos oferecem.

Como regra geral, este campo vem definido com 0x00_{hex} de fábrica, pois não possui qualquer tipo de configuração. No entanto, dispositivos como interruptores de toque podem já vir pré-configurados de fábrica. Todavia, essa situação não desobriga a uma reprogramação deste. No capítulo 4.2 é explicada uma técnica que permite identificar o grupo de uma forma automática.

Na tabela seguinte podem ser observados os diferentes valores que podem ser atribuídos ao campo de grupo. A tabela completa pode ser consultada no Anexo 3.

Tabela 9: Descrição dos valores associados ao campo *SRC Group*

SRC Group – Grupo de dispositivos de origem	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0x07	Reservado
0x08 a 0xF7	Grupo do dispositivo (consultar a Tabela 19)
0xF8 a 0xFD	Reservado
0xFE	Dispositivo de controlo de divisão (<i>Room</i>)
0xFF	Reservado

No caso de nos encontrarmos perante um dispositivo de controlo de *Room*, os campos relativos ao *Sector* e *Room* anteriores devem conter o valor específico da localização do controlador, encontrando-se neste campo de *Group* o valor 0xFE_{hex}. Esta situação será explicada posteriormente neste subcapítulo.

A Tabela 19, encontrada no Anexo 3, exhibe os grupos que podem ser definidos no campo de *Group* de origem. Note-se que apenas são indicados nessa tabela os grupos base, já que os subgrupos com as definições pormenorizadas podem ser encontrados nas tabelas seguintes do mesmo anexo.

SRC ID

O campo de ID limita-se à identificação do dispositivo. Partindo do princípio que existe a possibilidade da existência de dois dispositivos no mesmo local e com as mesmas características, este campo permite distingui-los através de um ID diferente.

Tabela 10: Descrição dos valores associados ao campo SRC ID

SRC ID – Identificação do dispositivo de origem	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01	Identificação do dispositivo (inclui a função de controlador)
0x02 a 0xFD	Identificação do dispositivo
0xFE	Dispositivo de controlo de grupo (<i>Group</i>)
0xFF	Reservado

No caso de ser pretendido incluir num dispositivo a função de controlador, tal como já foi apresentado no capítulo 3.5 , o *ID* deve ser definido como 0x01_{hex}.

Na eventualidade da presença de um dispositivo de controlo de *Group*, os campos relativos ao *Sector*, *Room* e *Group* anteriores devem conter o valor específico da localização e tipo do controlador, encontrando-se neste campo de *ID* o valor 0xFE_{hex}. Esta situação é explicada adiante neste subcapítulo.

Endereçamento de destino

Tal como ocorre no endereçamento de origem, também este é composto por 4 *bytes* divididos em quatro campos semelhantes, contendo apenas diferença no nome: são definidos por *DST* (*destination*) ao invés de *SRC* (*source*) e na funcionalidade de 0xFF, que se encontrava reservada para endereçamento de origem, mas que agora permite seleccionar todos os dispositivos do campo onde esse valor estiver inserido.

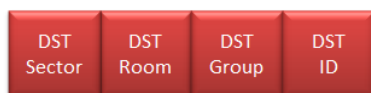


Figura 16: Campos de endereçamento de destino

DST Sector

Este campo é semelhante ao campo de *SRC Sector*. Cada dispositivo de destino tem a obrigação de atender ao seu número de *Sector* e ao valor 0xFF_{hex}, exceptuando-se o dispositivo de controlo geral, que apenas atende ao valor 0xFE_{hex}.

Tabela 11: Descrição dos valores associados ao campo *DST Sector*

DST Sector – Sector do dispositivo de destino	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0xFC	Sector do dispositivo
0xFD	Dispositivo móvel
0xFE	Dispositivo de controlo geral (<i>Global</i>)
0xFF	Selecciona todos os sectores

DST Room

Este campo é semelhante ao campo de *SRC Room*. Cada dispositivo de destino tem a obrigação de atender ao seu número de *Room* e ao valor 0xFF_{hex}, exceptuando-se o dispositivo de controlo de sector, que apenas atende ao valor 0xFE_{hex}.

Tabela 12: Descrição dos valores associados ao campo *DST Room*

DST Room – Divisão do dispositivo de destino	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0x07	Reservado
0x08 a 0xF7	Divisão do dispositivo (consultar o Anexo 2)
0xF8 a 0xFC	Reservado
0xFD	Dispositivo móvel
0xFE	Dispositivo de controlo de sector (<i>Sector</i>)
0xFF	Selecciona todas as divisões

DST Group

Este campo é semelhante ao campo de *SRC Group*. Cada dispositivo de destino tem a obrigação de atender ao seu número de *Group* e ao valor 0xFF_{hex}, exceptuando-se o dispositivo de controlo de divisão, que apenas atende ao valor 0xFE_{hex}.

Tabela 13: Descrição dos valores associados ao campo *DST Group*

DST Group – Grupo do dispositivo de destino	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01 a 0x07	Reservado
0x08 a 0xF7	Grupo do dispositivo (consultar a Tabela 19)
0xF8 a 0xFD	Reservado
0xFE	Dispositivo de controlo de divisão (<i>Room</i>)
0xFF	Selecciona todos os grupos

DST ID

Este campo é semelhante ao campo de *SRC ID*. Cada dispositivo de destino tem a obrigação de atender ao seu número de *ID* e ao valor 0xFF_{hex}, exceptuando-se o dispositivo de controlo de grupo, que apenas atende ao valor 0xFE_{hex}.

Tabela 14: Descrição dos valores associados ao campo *DST ID*

DST Group – Grupo do dispositivo de destino	
Valor (hex)	Descrição
0x00	Dispositivo sem configuração
0x01	Identificação do dispositivo (inclui a função de controlador)
0x02 a 0xFD	Identificação do dispositivo
0xFE	Dispositivo de controlo de grupo (<i>Group</i>)
0xFF	Selecciona todos os dispositivos (todos os ID's)

Endereçamento de dispositivos de controlo

Os dispositivos de controlo, tipicamente definidos pelos valores $0xFE_{hex}$ nas tabelas anteriores, possibilitam a inserção de elementos tais como servidores ou centrais de controlo em vários pontos do sistema. Embora esta característica possa não ter sido discutida da forma mais simples e clara nos pontos anteriores, a imagem seguinte pretende explicar de uma forma directa a diferença entre dispositivos de controlo global, de sector, de divisão e de grupo.

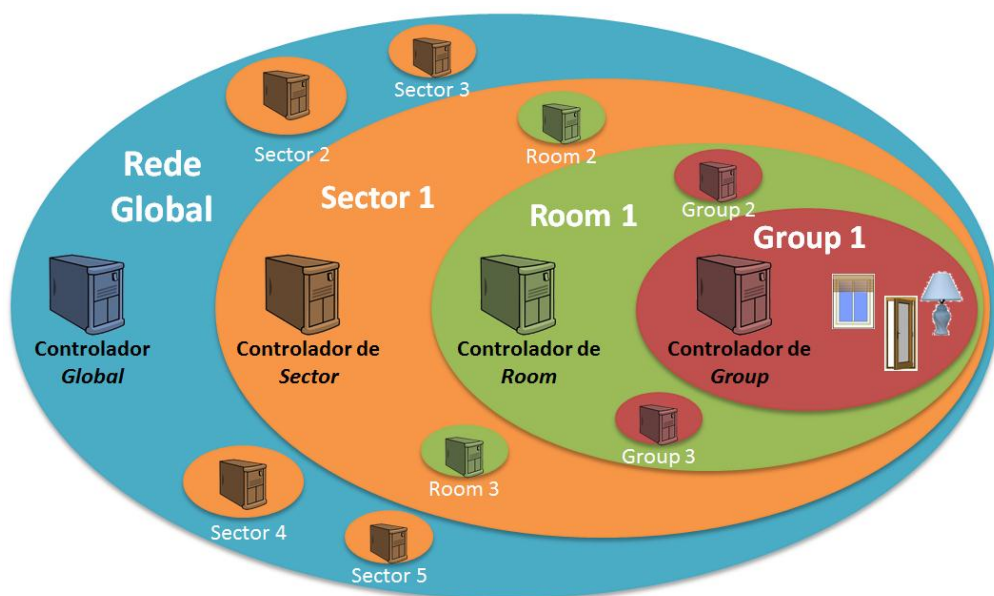


Figura 17: Diferença e localização de controladores globais, de sector, de divisão e grupo

A tabela seguinte mostra um exemplo de configuração de endereços para os vários controladores da Figura 17.

Tabela 15: Exemplo de configuração dos controladores da Figura 17

Controlador	Sector	Room	Group	ID	Exemplo
Global	0xFE	0xFE	0xFE	Qualquer	Controlador dentro de toda a rede
Sector	1	0xFE	0xFE	Qualquer	Controlador do 1º andar do prédio
Room	1	1	0xFE	Qualquer	Controlador da sala
Group	1	1	1	0xFE	Controlador da iluminação

3.6.4 Tipos de tramas

O protocolo B-Live Wireless define três grupos de tramas com diferentes funcionalidades: tramas de operações normais, tramas de diagnóstico e tramas de configuração. Estas diferentes tramas são definidas através de um carácter inserido no campo de *Frame type*. O tamanho dos dados que são inseridos até ao final da trama variam conforme o tipo de trama seleccionado.

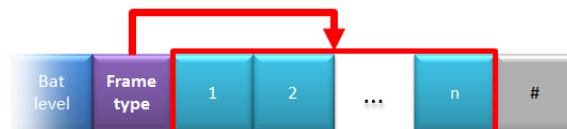


Figura 18: Os campos variáveis dependem do tipo de trama

Grupo de operação normal

Os pacotes referentes ao grupo de operação normal destinam-se à execução de tarefas comuns do dispositivo. São pacotes utilizados em mais de 90% do tempo de funcionamento. Dentro deste grupo, existem quatro tipos diferentes de pacotes: ordem de operação, *acknowledge*, informação de tempo-real e informação de localização.

Ordem de operação

Esta trama é definida pelo carácter ":". Destina-se à operação normal do dispositivo nas tarefas para o qual foi concebido. É utilizada quando é pretendido fazer um pedido de alteração de estado e, por essa razão, inclui campos relativos às entradas e saídas. A figura seguinte ilustra a estrutura de uma trama de operação.



Figura 19: Estrutura de uma trama de ordem de operação

De acordo com a figura anterior é possível observar que a trama contém três campos *OUT* e três campos *IN*. Estes referem-se às saídas e entradas físicas disponíveis nos módulos. Neste tipo de trama, os campos de *IN* são ignorados, visto não ser possível alterar o estado de uma entrada. Os campos de *OUT* podem ser atribuídos com valores entre $0x00_{hex}$ e $0xFF_{hex}$, que podem ser consultados na Tabela 36, no Anexo 4.

No caso em que o dispositivo a ser controlado não ofereça possibilidades de controlo de intensidade, deve ser utilizado um nível de *threshold* definido no *software* com a referência de $0x7F_{hex}$, de forma a diferenciar o estado ligado do estado desligado.

Na Figura 20 pode observar-se um exemplo prático onde esta trama é aplicada: o utilizador pressiona um botão e este envia uma trama de ordem de operação normal para a lâmpada.



Figura 20: Exemplo de uma trama de ordem de operação normal

Acknowledge

É uma resposta à trama de operação normal e é identificada pelo carácter “k” minúsculo (do inglês acKnowledge). Tem como principal objectivo informar o emissor que a mensagem foi recebida e as alterações foram efectuadas com sucesso. A figura seguinte ilustra a estrutura de uma trama de *acknowledge*.



Figura 21: Estrutura de uma trama de *acknowledge*

Na Figura 22 pode observar-se um exemplo prático onde esta trama é aplicada. Seguindo o exemplo da Figura 20, a lâmpada responde de volta ao interruptor com uma mensagem de *acknowledge*.



Figura 22: Exemplo de uma trama de *acknowledge*

Informação de tempo-real

Esta trama é identificada pelo carácter “R” (do inglês *Real-time*) e destina-se a ser utilizada por dispositivos que necessitem de comunicações em tempo-real. É uma trama destinada ao *gateway*. São anexados os campos data[1] ... data[n] de tamanho variável, que se referem às informações provenientes do dispositivo. A figura seguinte ilustra a estrutura de uma trama de *acknowledge*.



Figura 23: Estrutura de uma trama de tempo-real

Na Figura 24, pode observar-se um exemplo prático onde esta trama é aplicada: o utilizador dispõe de um monitor de sinais vitais, as informações recolhidas pelo monitor são enviadas em tempo real para o *gateway* e, posteriormente para o servidor.



Figura 24: Exemplo de uma trama de tempo-real

Informação de localização

Esta trama identifica-se com o carácter “L” e destina-se a fornecer informações ao servidor sobre a localização de um dispositivo móvel numa habitação.

O servidor deverá possuir um mapa da habitação e um algoritmo que efectue os cálculos da localização do utilizador, através dos níveis de potência da mensagem do dispositivo móvel definida como nível RSSI, quando esta é recebida pelos controladores de divisão (*Room*). A figura seguinte pretende ilustrar o que foi descrito atrás.



Figura 25: Variação do nível de RSSI com a distância ao emissor

Desta forma, os controladores de divisão enviam esta mensagem para o *gateway*, que, por sua vez, se encarregará de a fazer chegar ao servidor, contendo os campos que identificam o dispositivo: o endereço, o *Sequence Number* original e o nível de potência RSSI com que a trama foi originalmente recebida. A figura seguinte ilustra a estrutura de uma trama de localização.



Figura 26: Estrutura de uma trama de localização

Na Figura 27, pode observar-se um exemplo prático onde esta trama é aplicada. O utilizador dispõe de uma *tag* que envia uma mensagem para o ambiente envolvente. Esta é recebida pelos controladores que se encontram dentro do raio de alcance e reencaminhada para o *gateway* que, por sua vez, a reencaminha para o servidor.

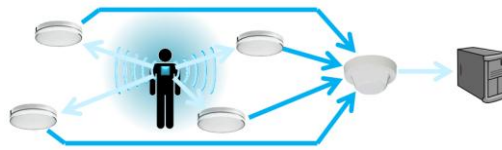


Figura 27: Exemplo da operação de localização

Grupo de diagnóstico

Os pacotes referentes ao grupo de diagnóstico são destinados à obtenção de informações acerca dos dispositivos. Dentro deste grupo existem cinco tipos diferentes de tramas: pedido *"Are you alive?"*, informação *"I'm alive!"*, pedido de *status*, informação de *status* e informação de erro.

Pedido "Are you alive?"

Esta trama é identificada pelo carácter "?" e destina-se a ser utilizada para questionar um dispositivo de forma a verificar se este existe e se encontra activo. Pode ser utilizada também para pesquisar dispositivos novos sem configuração. O dispositivo que envia esta questão deve aguardar por uma trama de informação *"I'm alive!"* como resposta. A Figura 28 seguinte ilustra a estrutura de uma trama *"Are you alive?"*.

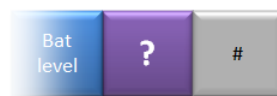


Figura 28: Estrutura de uma trama de pedido *"Are you alive?"*

Na Figura 29, é possível observar um exemplo prático desta trama. O dispositivo controlador da sala de estar envia um pedido *"Are you alive?"* a todos os dispositivos na sala, de forma a verificar se todos se encontram em funcionamento.



Figura 29: Exemplo da operação de pedido *"Are you alive?"*

Informação "I'm alive!"

Esta trama é identificada pelo carácter "!" e destina-se a informar o sistema que o dispositivo se encontra activo. Os dispositivos devem enviar esta trama após terem recebido um

pedido "Are you alive?", aguardando um tempo aleatório de forma a evitar possíveis colisões de dados. A Figura 30 seguinte ilustra a estrutura de uma trama "I'm alive!".



Figura 30: Estrutura de uma trama de informação "I'm alive!"

Na Figura 31 é possível observar um exemplo prático deste pedido. Na fase programação de um novo dispositivo, o interruptor envia continuamente o pedido "Are you alive?". O candeeiro vai, então, responder com a trama "I'm alive!", dando assim início à fase de programação de ambos.



Figura 31: Exemplo da utilização da trama "I'm alive!"

Pedido de status

Esta trama é identificada pelo carácter "S" (do inglês: *Status*) e destina-se a pedir o estado de carga da bateria, entradas e saídas do dispositivo a que se destina. A resposta a este pedido é a trama de informação de *status*, descrita no ponto seguinte. A Figura 32 ilustra a estrutura de uma trama de pedido de *status*.



Figura 32: Estrutura de uma trama de pedido de status

Na Figura 33 é possível observar um exemplo prático deste pedido. O controlador do quarto questiona o candeeiro acerca do seu estado: ligado ou desligado.



Figura 33: Exemplo da utilização da trama de pedido de status

Informação de status

Esta trama é, tal como a trama de pedido de *status*, identificada pelo carácter "S". A única diferença entre as duas reside no facto de esta conter dados anexados, o que não acontece na trama anterior.

Destina-se a responder ao pedido de *status* enviando as informações relativas ao estado actual das saídas (campos OUT) e das entradas (campos IN). A Figura 34 ilustra a estrutura de uma trama de pedido de *status*.



Figura 34: Estrutura de uma trama de informação de *status*

Na Figura 35 é possível observar um exemplo prático desta trama. De acordo com o exemplo, e seguindo o processo mencionado na trama de pedido de *status*, o candeeiro responde a esse pedido com esta trama de informação de estado.



Figura 35: Exemplo da utilização da trama de informação de *status*

Informação de erros

Esta trama é identificada pelo carácter "E" (do inglês: *Error*) e destina-se a informar qualquer dispositivo de controlo da existência de um erro com o dispositivo que originou a mensagem, ou com outro dispositivo com o qual foram detectados problemas. A Figura 36 mostra a estrutura de uma trama de erro.

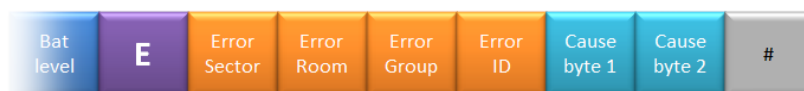


Figura 36: Estrutura de uma trama de erro

Na trama é incluído o endereço completo do dispositivo onde foi detectado o problema, assim como 2 bytes que referenciam a causa deste. Ao contrário do que acontece com os outros campos do protocolo, aqui o erro não é determinado através de um índice numérico, mas determinado *bit a bit*. Isto significa que cada *bit* dos 2 *bytes* refere um erro específico, havendo a possibilidade de existência de vários erros em simultâneo. As causas de erro podem ser consultadas na Tabela 37, disponível no Anexo 4.

Na Figura 37 é possível observar um exemplo prático da utilização desta trama. O controlador da sala tenta comunicar com o candeeiro, no entanto a comunicação não é bem-

sucedida, pelo que o controlador vai informar o *gateway* e, posteriormente, o servidor da existência de um problema com o cadeeiro.

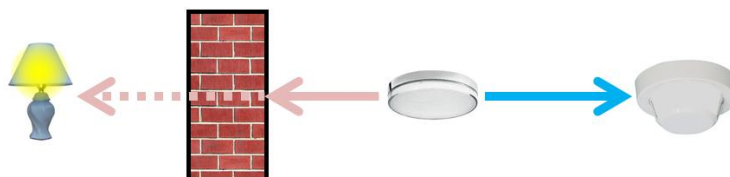


Figura 37: Exemplo da utilização da trama de erro por outro dispositivo

Apesar de, no exemplo anterior, não ter sido o próprio dispositivo com problemas a reportar o erro, existem situações em que este pode desempenhar essa tarefa, como é apresentado na Figura 38, no qual o cadeeiro comunica com controlador enviando uma mensagem de erro a informar que este se encontra em sobreaquecimento.



Figura 38: Exemplo da utilização da trama de erro pelo próprio dispositivo

Grupo de configuração

Os pacotes referentes ao grupo de configuração são utilizados de forma a poder configurar parâmetros de cada dispositivo como potência, canal de comunicação, endereços, entre outros. Fazem parte deste grupo três tipos diferentes de tramas: operação de configuração, alteração de endereço e *acknowledge* de configuração.

Operação de configuração

Esta trama é identificada pelo carácter "*C*" (do inglês: *Configuration*) e é utilizada de forma a configurar alguns parâmetros do dispositivo a que se destina. Estes podem variar de dispositivo para dispositivo. No entanto existem dois campos que são comuns em qualquer elemento: o campo *Ch select* e o campo *TX Power*, tal como pode ser observado na Figura 39. Dos seis campos, dois são utilizados e quatro estão livres para futuras implementações.



Figura 39: Estrutura de uma trama de configuração

O campo *Ch select* é um campo empregue para definir qual o canal em que o dispositivo vai operar e, por defeito, é utilizado o canal 11. O valor deste campo pode variar desde 11 até 26.

O campo *TX Power* é utilizado para definir a potência de transmissão do módulo. Nem sempre é necessário utilizar a potência máxima, pois no caso em que o dispositivo se encontra a operar com uma bateria, o tempo de vida útil desta será mais reduzido. Por esta razão, é permitido definir a potência de cada dispositivo, segundo a Tabela 38 disponível no Anexo 4.

Na Figura 40 é possível observar um exemplo prático onde esta trama é empregue. Após a sua instalação, o candeeiro tem de ser configurado antes de poder ser utilizado no sistema. Definições como a potência e o canal são as primeiras a ser configuradas.



Figura 40: Exemplo da operação de configuração

Operação de alteração de endereços

Esta trama é identificada pelo carácter "A" (do inglês: *Address*) e é utilizada de forma a configurar os endereços registados na memória do dispositivo. Estes dispõem de um endereço de origem e de uma tabela de destino que pode ser configurada de forma a permitir vários destinatários. A estrutura da trama pode ser observada na Figura 41.



Figura 41: Estrutura de uma trama de configuração

Existem quatro campos referentes ao novo endereço de origem que o dispositivo vai tomar (*NEW SRC x*), um campo de operação (*Oper.*) e mais quatro campos, desta vez referentes ao endereço de destino para o qual o dispositivo irá enviar as mensagens (*NEW DST x*).

Os campos de endereçamento funcionam de forma semelhante aos encontrados no início da trama, que podem ser consultados na Figura 14: Estrutura de uma trama genérica. Para retomar o dispositivo ao estado de fábrica, os campos relativos ao novo endereço de origem devem ser colocados todos a 0x00_{hex}.

O campo de operação, representado na tabela seguinte, define de que forma o endereço de destino é armazenado numa tabela existente na memória do módulo.

Tabela 16: Valores associados ao campo *Oper.* da trama de alteração de endereços

<i>Oper.</i> – Operação a efectuar relativamente ao endereço de destino		
Carácter	Descrição	Tamanho da tabela
>	Apaga a tabela e depois adiciona o endereço de destino referido	$t = 0$, $t++$
+	Adiciona à tabela o endereço de destino referido	$t++$
-	Remove da tabela o endereço de destino referido	$t--$
<i>outro</i>	Sem qualquer alteração	t

Na Figura 42 é possível observar um exemplo prático onde esta trama é empregue Após ser instalado, o candeeiro necessita de ser configurado antes de poder ser utilizado no sistema. Depois da potência e o canal terem sido configurados, podem agora ser configurados os endereços correctos.



Figura 42: Exemplo de uma trama de *alteração de endereços*

Acknowledge de configuração

A trama de *acknowledge* de configuração é identificada com um carácter "K" maiúsculo (do inglês: *acKnowledge*) e é uma resposta à mensagem de configuração do dispositivo e de configuração de endereços. O objectivo desta trama é de informar o dispositivo que anteriormente enviou as configurações e de que estas foram recebidas e guardadas com sucesso. Desta forma, a estrutura da trama de *acknowledge* é semelhante à mensagem de configuração anterior, exceptuando-se o carácter que identifica o tipo de trama, tal como pode ser observado na Figura 43 e na Figura 44.



Figura 43: Estrutura de uma trama de *acknowledge de configuração de dispositivo*



Figura 44: Estrutura de uma trama de *acknowledge de alteração de endereços*

Os campos acima representados a translúcido não são uma cópia exacta dos dados recebidos na mensagem anterior, mas antes os dados lidos da memória após a configuração anterior. Desta forma, é possível detectar eventuais erros durante o processo de configuração.

Na Figura 45, pode observar-se um exemplo prático onde esta trama é aplicada. Tal como acontece na operação de configuração, após a instalação, o candeeiro é configurado. Deste modo ele envia uma mensagem de *acknowledge* de configuração de volta ao programador, indicando o (in)sucesso dessa tarefa.

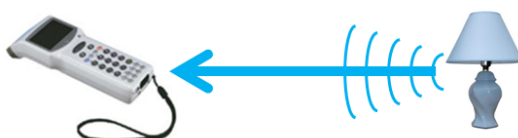


Figura 45: Exemplo de uma trama de *acknowledge* de configuração

3.6.5 Funcionamento

De forma a descrever o funcionamento do sistema este capítulo está dividido em três fases. A primeira, expõe a forma de adicionar e configurar novos dispositivos à rede; a segunda explica o processamento das comunicações entre dispositivos para diferentes casos e, finalmente, na última fase, é descrita a implementação das comunicações sobre o protocolo WFTT, já introduzido no início deste capítulo 3.

Configuração de dispositivos

De forma a configurar os módulos e os dispositivos, é necessária a existência de um programador, acessível às pessoas responsáveis pela instalação do sistema, para que seja possibilitada a configuração de todos os elementos do sistema no momento da instalação.

O programador pode ser um pequeno aparelho com ligação USB ou Bluetooth ou um dispositivo portátil com um visor e teclado.



Figura 46: Diferentes tipos possíveis de programadores

O programador deve permitir ao instalador seleccionar opções como o endereçamento (*Sector, Area, Group* e *ID*), a potência do sinal e o canal de transmissão, de forma a programar cada um dos módulos na instalação de uma forma simples. Estas opções devem surgir no visor em formato de texto e não em formato de código, sendo esse processamento transparente para o instalador. A Figura 47 ilustra uma ideia simples, possível de ser mostrada no ecrã do programador.



Figura 47: Exemplo de informação no programador portátil

A configuração de dispositivos, novos ou já em operação, resume-se em três simples passos. De notar que este método de programação é válido para todos os dispositivos.

- I. Definir as configurações: o instalador define as configurações necessárias no programador;



Figura 48: Configuração de um dispositivo - passo I

- II. Colocar o dispositivo no modo de programação: cada módulo dispõe de um botão de configuração que deve ser pressionado durante alguns segundos;



Figura 49: Configuração de um dispositivo - passo II

- III. Programar o dispositivo: o instalador pressiona o botão de programação e a configuração é enviada do programador para o dispositivo destinado.

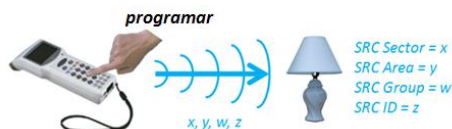


Figura 50: Configuração de um dispositivo - passo III

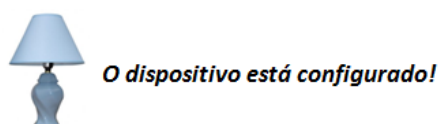


Figura 51: Configuração de um dispositivo – completa

Comunicação entre dispositivos

De forma a descrever o princípio de comunicação entre dispositivos, é exposto um exemplo baseado numa habitação comum, onde constam os seguintes dispositivos:

1. localizador *tag*;
2. monitor de sinais vitais;
3. detector de fumo;
4. actuador de estores;
5. sensor de estado da porta;
6. sensor de presença humana;
7. interruptor;
8. candeeiro do quarto;
9. candeeiro da casa de banho (a operar com a função de controlador);
10. candeeiro da cozinha (a operar com a função de controlador);
11. tomada de aquecimento (a operar com a função de controlador);
12. *gateway* (com a função de *Master* incorporada);
13. servidor.

É assumido que todos estes dispositivos já se encontram configurados e a operar normalmente. Os procedimentos para configurar a rede encontram-se no capítulo anterior.

A planta da habitação da Figura 52 pretende ilustrar o local onde cada dispositivo se encontra localizado. De notar que o localizador *tag* (1) e o monitor de sinais vitais (2) partilham do mesmo hardware, mas são explicados separadamente para maior simplicidade.

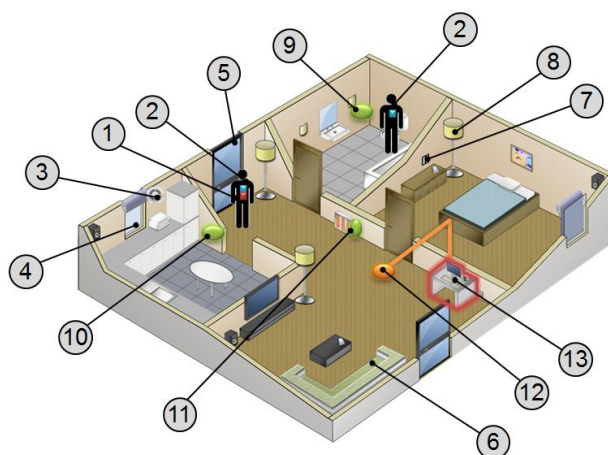


Figura 52: Um exemplo de um cenário onde o sistema se encontra aplicado

Seguidamente, são apresentadas descrições passo a passo para vários cenários de actuação e monitorização possíveis de existir dentro da habitação exibida anteriormente.

Controlo de iluminação

- I. O utilizador pressiona o interruptor (7) localizado na parede;



Figura 53: Exemplo de controlo de iluminação – passo I

- II. O interruptor envia uma trama contendo a informação de alteração de estado destinada ao candeeiro (8);



Figura 54: Exemplo de controlo de iluminação – passo II

- III. O candeeiro recebe a trama e efectua a acção pedida;



Figura 55: Exemplo de controlo de iluminação – passo III

- IV. Um pacote de *acknowledge* é enviado de novo para o interruptor (7) de forma a informar que a tarefa foi realizada com êxito;



Figura 56: Exemplo de controlo de iluminação – passo IV

- V. No mesmo instante do ponto III, o dispositivo de controlo (9) localizado perto dessa área também recebe a trama, enviando um registo para o *gateway* (12);



Figura 57: Exemplo de controlo de iluminação – passo V

- VI. O *gateway* recebe o registo e reencaminha para o servidor (13) através de *Ethernet* ou *Wi-Fi*, onde os dados são armazenados numa base de dados.



Figura 58: Exemplo de controlo de iluminação – passo VI

A sequência de eventos explicada é aplicada aos restantes actuadores e sensores, tais como: controlo de ventoinha, sensor de temperatura, controlo de tomadas ou de estores. Na situação em que a mensagem necessita de comunicar para um dispositivo que se encontra fora da sua área de cobertura, os controladores devem proporcionar o reencaminhamento dessa mensagem até ao destino.

Localização

- I. A *tag* de localização (1) envia uma trama periódica a cada poucos segundos;



Figura 59: Exemplo de localização – passo I

- II. Todos os controladores de divisão que se encontram dentro do raio de cobertura da *tag* (9, 10 e 11) recebem a trama e calculam o seu nível de potência;



Figura 60: Exemplo de localização – passo II

- III. Os controladores reúnem este valor e o endereço da *tag*, enviando-os dentro de uma trama para o *gateway* (12);



Figura 61: Exemplo de localização – passo III

- IV. O *gateway* recebe o registo e reencaminha para o servidor (13) através de *Ethernet* ou *Wi-Fi*, onde os dados são armazenados numa base de dados;



Figura 62: Exemplo de localização – passo IV

- V. O servidor reúne os vários dados e processa-os a fim de obter a localização do utilizador dentro da habitação.

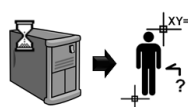


Figura 63: Exemplo de localização – passo V

Detecção de eventos

- I. O sensor localizado na porta (5) detecta que a porta foi aberta ou fechada manualmente e envia esta informação para o controlador (11). O controlador por sua vez reencaminha esta mensagem para o *gateway* (12);



Figura 64: Exemplo de detecção de eventos – passo I

- II. O *gateway* recebe esta informação e reencaminha-a para o servidor (13) através de *Ethernet* ou *Wi-Fi*, onde os dados são armazenados numa base de dados.



Figura 65: Exemplo de detecção de eventos – passo II

Detecção de presença

- I. O sensor localizado no sofá (6) detecta quando o utilizador se senta ou se levanta, enviando esta informação para o controlador da divisão onde se situa (11). O controlador reencaminha-a para o *gateway* (12);

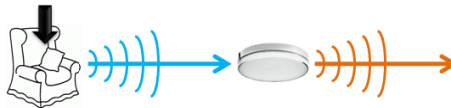


Figura 66: Exemplo de detecção de presença – passo I

- II. O *gateway* recebe esta informação e reencaminha-a para o servidor (13) através de *Ethernet* ou *Wi-Fi*, onde os dados são armazenados numa base de dados.



Figura 67: Exemplo de detecção de presença – passo II

Alarme

O detector localizado na cozinha (3) detecta a presença de fumo e envia uma mensagem imediatamente para o *gateway* (12). Neste caso, a mensagem não é reencaminhada pelos controladores (9, 10 e 11), embora estes também recebam, igualmente, a informação. O *gateway* reencaminha o alerta para o servidor (13).



Figura 68: Exemplo de detecção de alerta e alarme

Funcionamento global do sistema

Nesta fase, será descrito o funcionamento global do sistema englobando a tecnologia *WFTT*. De forma a elucidar esta sequência, deve fazer-se referência aos dispositivos existentes na Figura 52, assim como à respectiva descrição da operação.

Por defeito, todos os dispositivos da habitação que operam com baterias (1, 2 e 7) encontram-se no modo de baixo consumo (*sleep*). Como exemplos destes dispositivos podem considerar-se interruptores, monitorizadores de sinais vitais, localizadores *tag*, entre outros.

- I. A comunicação inicia-se com o *gateway* (12) a operar como *Master*, transmitindo o início da *superframe* do protocolo *WFTT*, constituída primeiramente pela sequência de ruído *Black-Burst*. Esta sequência garante que os restantes dispositivos suspendem as suas comunicações. Os dispositivos operados a bateria (1, 2 e 7) acordam durante a recepção desta *superframe*;

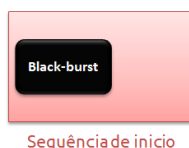


Figura 69: Funcionamento global do sistema com o protocolo *WFTT* – passo I

- II. O *Master* transmite o *trigger packet* em *broadcast* para todos os dispositivos da habitação. Este pacote contém todas as informações relativas a prioridades, tempos da janela protegida (*PW*) e de contenção (*CW*). Terminando esta transmissão, é iniciada a janela protegida;



Figura 70: Funcionamento global do sistema com o protocolo WFTT – passo II

- III. Todos os dispositivos que necessitem de efectuar comunicações carecem da obrigação de cumprimento do protocolo WFTT. Desta forma, as prioridades são organizadas e todos os dispositivos que necessitam de aguardar um intervalo de tempo até poderem efectuar as suas comunicações, são colocados em modo de *sleep*. No exemplo, assume-se que os dispositivos com maior prioridade são os monitorizadores de sinais vitais (2), sendo os primeiros a comunicar segundo a ordem definida pelo *Master*;



Figura 71: Funcionamento global do sistema com o protocolo WFTT – passo III

- IV. A janela protegida reserva, igualmente, espaço para outras comunicações de alta prioridade, como dispositivos controladores. Estes utilizam esta janela para transmitir informações como a localização das *tags* dentro da habitação ou mensagens de registo de eventos. Os controladores (9, 10 e 11), que se encontravam em modo *sleep*, acordam em sequência para efectuar as suas comunicações. Na figura seguinte pode ser observada a calendarização das transmissões por parte dos controladores;



Figura 72: Funcionamento global do sistema com o protocolo WFTT – passo IV

- V. O último campo é reservado a situações de emergência, por exemplo no caso de ter sido detectado uma fuga de gás, um incêndio ou a existência de fumo. Relativamente ao exemplo apresentado, na ocorrência da detecção de fumo na cozinha, o detector iria efectuar a sua comunicação de emergência neste campo, tal como é demonstrado na figura seguinte. Importa referir que, mesmo que não exista qualquer tipo de emergência, devem ser sempre efectuadas comunicações de forma a manter o canal ocupado;



Figura 73: Funcionamento global do sistema com o protocolo WFTT – passo V

- VI. Quando o espaço da *superframe* reservado para emergência termina, a janela protegida é fechada, abrindo-se, agora, a janela de contenção. Tal como foi descrito no protocolo WFTT no capítulo 3.2, esta janela permite comunicações de dispositivos de baixa prioridade segundo o mecanismo CSMA/CA. Os dispositivos que se encontravam em modo *sleep*, acordam no início desta janela e tentam efectuar as suas comunicações segundo esse mecanismo. Consequentemente, o tempo de início de comunicação de cada dispositivo é desconhecido.



Figura 74: Funcionamento global do sistema com o protocolo WFTT – passo VI

Quando a janela de contenção termina, todas as comunicações são suspensas e um novo ciclo é iniciado, dando início a uma nova *superframe*. Os dados que não foram transmitidos são agendados para o próximo ciclo elementar. Um exemplo dessa ocorrência é relativo às *tags* de localização (1), onde os controladores (9, 10 e 11) recebem a trama periódica mas apenas têm oportunidade de informar o *gateway* (12) e o servidor (13) no próximo ciclo elementar, nos seus campos respectivos da janela protegida.

4. Desenvolvimento

4.1 Software

Como base de programação, optou-se pela utilização de famílias de microcontroladores PIC® da Microchip® [13], devido à actual existência de placas de desenvolvimento com esses dispositivos. De forma a criar e compilar o *software* foi utilizado o ambiente de desenvolvimento MPLAB X [14] da Microchip®. Para mais informações é possível consultar o Anexo 5 onde é apresentado um tutorial de configuração de todo o ambiente utilizado.



Figura 75: O logótipo da Microchip® e do MPLABX®

4.1.1 Introdução

O *software* do sistema B-Live Wireless é constituído por diferentes bibliotecas desenvolvidas numa abordagem *bottom-up*, isto é, começando no nível do *drive* de *hardware* e subindo até ao nível do programa principal. Estas bibliotecas têm como objectivo simplificar a utilização do sistema, oferecendo, ao mesmo tempo, alguma transparência de utilização.

A Figura 76 representa o esquema de bibliotecas existentes no B-Live Wireless. Como pode ser observado, a cor laranja encontram-se as bibliotecas de baixo nível. Depois à medida que as camadas são adicionadas, surgem as bibliotecas referentes ao protocolo IEEE802.15.4. Já no topo encontram-se as bibliotecas pertencentes à implementação da versão alterada do protocolo WFTT e ao B-Live, onde são incluídas as bibliotecas extra referentes aos diferentes grupos de dispositivos.

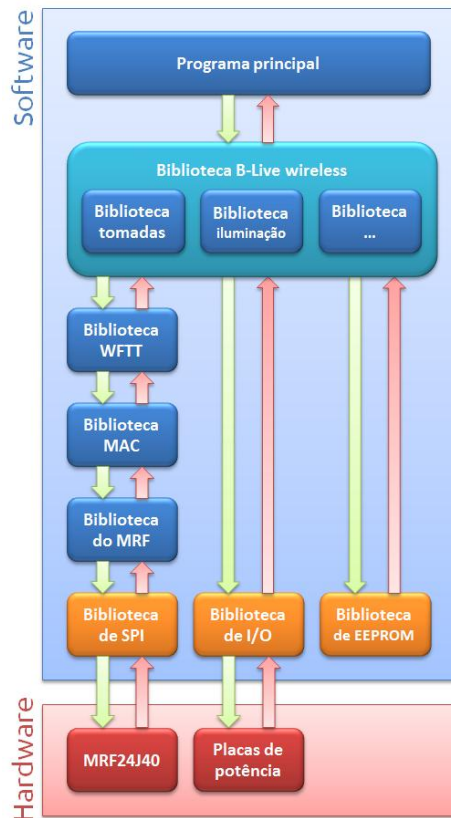


Figura 76: Esquema de bibliotecas existentes no B-Live Wireless

De forma a simplificar a criação e compilação do código, todos os módulos existentes no sistema irão conter o mesmo código, sendo a selecção da funcionalidade efectuada após a primeira inicialização e configuração deste.

4.1.2 Bibliotecas

As placas de desenvolvimento utilizadas já continham um *software* desenvolvido que foi modificado de acordo com as necessidades deste projecto. Nele constavam as seguintes bibliotecas:

- Biblioteca de EEPROM
- Biblioteca de I/O
- Biblioteca de SPI
- Biblioteca de suporte ao MRF
- Biblioteca da camada MAC
- Biblioteca do protocolo WFTT (excepção: não desenvolvido até à data)

Biblioteca de EEPROM

A biblioteca de EEPROM foi desenvolvida pela Microchip® e já se encontrava incluída no ambiente de desenvolvimento. Esta oferece funcionalidades referentes à operação com a memória não volátil interna do microcontrolador. De entre as funções existentes nesta biblioteca destacam-se fundamentalmente a operação de escrita (*write*) e de leitura (*read*). No B-Live *wireless*, estas funções desempenham papéis cruciais pois permitem que todas as configurações dos dispositivos sejam mantidas após um *reset* ou falha na alimentação.

Biblioteca de I/O

A biblioteca de I/O contém todas as configurações de *hardware* externo existente nas placas (o esquemático desta placa pode ser consultado no Anexo 6). A biblioteca foi alterada de acordo com as exigências deste projecto, que podem ser consultadas no capítulo 4.2 .

Biblioteca de SPI

Esta biblioteca contém todas as configurações relativas ao módulo de SPI interno do microcontrolador, introduzindo funções para receber ou enviar dados por esta porta. A biblioteca foi mantida intacta por não necessitar de qualquer alteração.

Biblioteca de suporte ao MRF

A biblioteca do MRF oferece todas as funções necessárias à comunicação entre o microcontrolador e o módulo MRF24J40, tais como a escrita e a leitura de endereços de memória ou comandos de configuração. Tal como no caso anterior, também esta não necessitou de qualquer alteração.

Biblioteca da camada MAC

As funções referentes à camada MAC do protocolo IEEE802.15.4 implementam toda a rede sobre o módulo MRF. Desta biblioteca constam funções como a criação e leitura de pacotes, preenchimento de endereços e dados, configuração de interrupções, potência, canais, entre outras características e envio de comandos.

Esta biblioteca necessitou, no entanto, de alterações para que o método de segurança pudesse ser implementado. Consequentemente, é possível activar ou desactivar a encriptação de dados.

Biblioteca do protocolo WFTT

A biblioteca referente à versão alterada do protocolo WFTT não se encontrava previamente desenvolvida, pelo que teve de ser criada de raiz. Nesta biblioteca são incluídas funções que permitem o cumprimento do protocolo em relação a instantes de tempo utilizados para comunicar e para a inactividade do dispositivo, tais como a leitura do *trigger packet*, o ajuste dos temporizadores para permitir acordar o dispositivo no momento exacto. Este processo é transparente para as camadas superiores.

4.1.3 B-Live

O B-Live *wireless* é composto por várias bibliotecas, cada uma referente a diferentes funcionalidades. As bibliotecas estão relacionadas com os grupos definidos no capítulo 3.6.3 e encontram-se expostas na Tabela 17.

Tabela 17: As bibliotecas existentes no B-Live Wireless

Bibliotecas incluídas no B-Live wireless
Iluminação
Portas / Janelas
Estores
Tomadas
Autoclismo
Torneiras
Menu vocal
Intercomunicadores
Comando multimédia
Dispensadores
Ventilação / Aquecimento
Interruptores
Dispositivos móveis
Elevador
Sensores

Estas bibliotecas criam o interface entre o programa principal e os restantes elementos descritos anteriormente, observáveis na Figura 76. Cada uma delas contém as definições e funções necessárias à operação do módulo em questão.

Podem ser sublinhados alguns aspectos que diferenciam estas bibliotecas:

- I. A biblioteca de iluminação abrange funções que permitem controlar o estado de qualquer lâmpada, *spot* ou candeeiro;
- II. A biblioteca de portas e janelas contém funções que controlam a abertura e fecho de portas ou janelas, imediatamente ou mediante um temporizador;
- III. A biblioteca de estores tem funções que configuram as saídas de acordo com o protocolo a ser utilizado no mecanismo dos estores;
- IV. A biblioteca de tomadas contém apenas as funções comuns para as ligar ou desligar;
- V. A biblioteca para o controlo do autoclismo configura as saídas de forma a oferecerem compatibilidade com um mecanismo de controlo deste;
- VI. A biblioteca de menu vocal ainda não se encontra implementada, mas irá envolver funções e definições relativas a frases ou questões que serão posteriormente convertidas em comandos de voz;
- VII. A biblioteca de intercomunicadores também ainda não se encontra implementada. No entanto irá incluir funções relativas a operações com porteiros de voz, de chave electrónica ou de cartões sem fios;
- VIII. A biblioteca de comando multimédia também não se encontra implementada. Irá englobar funções e definições relativas a um comando de infravermelhos com tecnologia *point-and-play*, no qual o utilizador aponta o comando para um dispositivo da casa e pressiona o botão de acção, alterando assim o seu estado;
- IX. A biblioteca de dispensadores engloba essencialmente as funções e o registo dos elementos disponíveis no dispensador (localização, quantidade, entre outros);
- X. A biblioteca de ventilação e de aquecimento inclui funções de controlo e variáveis de registo de temperaturas permitindo controlar um aparelho de ar condicionado ou um aquecedor, por exemplo;
- XI. A biblioteca de interruptores apenas dispõe de funções especializadas a ler as entradas, aplicando métodos de filtragem de ruído por *software*;
- XII. A biblioteca de dispositivos móveis é uma biblioteca que, essencialmente, inclui tabelas de registo relativas às características das redes onde estes dispositivos se encontram. Engloba, também, funções para criar, compactar e colocar em lista de espera os pacotes de tempo-real;
- XIII. A biblioteca de elevador aglomera funções e configurações relativas ao protocolo de operação dos elevadores, disponibilizando para o exterior pedidos de chamada ou movimentação do elevador;

- XIV. A biblioteca de sensores contém todas as funções, variáveis, tabelas, definições e configurações associadas a qualquer um dos diferentes tipos de sensores. Incluem-se, por exemplo, pedidos de leitura de temperatura, de humidade, de gás, entre outros. Esta biblioteca utiliza alguns pinos de entrada analógica do microcontrolador para realizar amostragem de dados reais provenientes de sensores que assim o exigiam.

O programa principal selecciona qual a biblioteca na qual vai operar após a inicialização e a configuração do módulo por parte do instalador.

No capítulo seguinte, serão expostas as razões que favorecem a necessidade de incluir todas as bibliotecas e funcionalidades num *software* único.

4.2 Hardware

De forma a testar o *software* do sistema B-Live Wireless, procedeu-se à utilização de uma placa de desenvolvimento existente. Esta placa, cuja fotografia é mostrada em seguida, está equipada com um PIC18F26K20, com memória de 64KB, uma capacidade de processamento de 16MIPS a uma velocidade de 64MHz, e um módulo sem fios MRF24J40MA baseado no *standard* IEEE802.15.4.

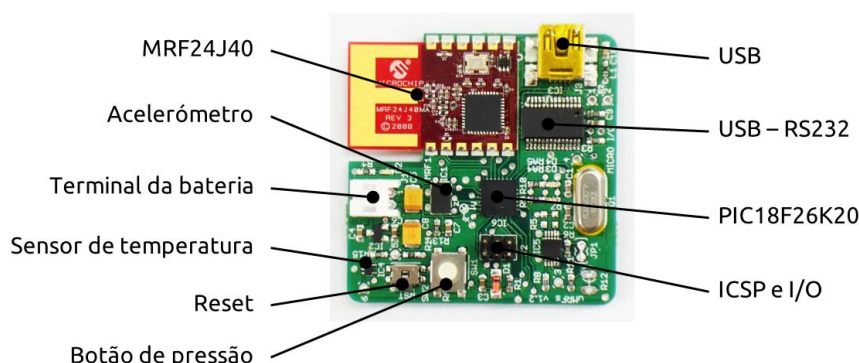


Figura 77: Placa de desenvolvimento utilizada

No Anexo 6, pode ser observado o esquema da placa de circuito impresso. Embora esta placa contenha uma grande parte dos dispositivos essenciais ao desenvolvimento deste projecto, não se destina a ser utilizada como um produto final. Desta forma, é necessário investigar e desenvolver uma solução que se adapte ao objectivo pretendido.

4.2.1 Introdução

Para o dimensionamento de placas de circuito impresso é necessário pensar sempre numa frente económica, simples e que permita uma expansão a desenvolvimentos posteriores. Sendo este um dos objectivos deste projecto, optou-se por dividir os dispositivos em dois

elementos: um de controlo e um de potência. A intenção de separar os elementos recai sobre o facto de ser desnecessário construir placas dimensionadas de forma diferente para cada fim a que se destinam. Desta forma, o processamento é feito na placa de controlo, a placa de potência destina-se apenas a fornecer a alimentação e os componentes externos necessários.

4.2.2 Controlo

A placa de controlo é o elemento principal de todo o sistema. Contém o microcontrolador e o transceptor sem fios responsável pelas comunicações.

A placa é constituída por os seguintes dispositivos principais:

- microcontrolador PIC18F26K20;
- módulo MRF24J40;
- bateria de polímeros de lítio (Li-Po) de 3.7V;
- gestor da bateria;
- sensor de temperatura.

Os seguintes sinais estão disponíveis para o exterior através de um conector:

- reset;
- pinos de programação ICSP;
- comunicação série (RS232-TTL e SPI);
- pinos de entrada (digital, analógica e de interrupção) e saída;

É importante que o tamanho seja reduzido para que, dessa forma, se consiga aproveitar o número máximo de placas durante o processo de fabrico em massa.

A Figura 78 mostra o diagrama referente às ligações principais da placa.

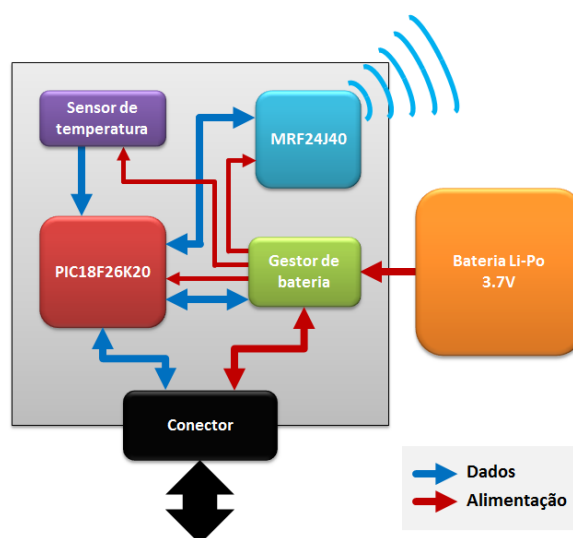


Figura 78: Diagrama de blocos da placa de controlo

Na Figura 79 pode ser observado um traçado final mostrando o posicionamento previsto dos componentes na placa de circuito impresso.

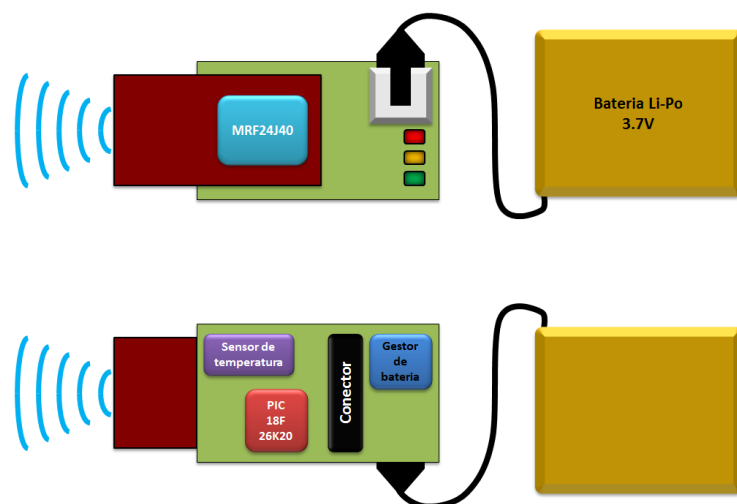


Figura 79: Simulação da organização possível para as duas faces da placa de controlo

4.2.3 Potência

A placa de potência revela a sua importância ao realizar a ponte entre o dispositivo de controlo e o dispositivo controlado. Esta placa é a responsável por fornecer à placa de controlo a alimentação devida, a filtragem de entradas e o isolamento eléctrico das saídas.

A placa de potência varia conforme a funcionalidade a que se destina. No entanto, é possível mencionar dois tipos importantes destas: as de saída, que contêm relés, TRIAC's ou outros elementos de potência, e as de entrada, que contêm filtros de entrada.

Uma placa de potência é constituída por os seguintes dispositivos principais:

- fonte de alimentação de 5V e respectivos circuitos de protecção;
- relés ou TRIAC's de potência (no caso de placas de saída de potência);
- circuito de filtragem de entradas (no caso de placas de entrada);
- qualquer outro elemento necessário à funcionalidade a que se destina.

Os seguintes sinais estão disponíveis para o exterior através de conectores:

- pinos de entrada e saída;
- conector de alimentação de entrada (220V~);
- conector de saída para dispositivos externos (para placas de saída);
- conector de entrada para dispositivos externos (para placas de entrada).

Tal como acontece com as placas de controlo, também aqui é importante que o tamanho seja o mais reduzido possível. Não obstante, o formato físico da placa deve ajustar-se ao encapsulamento ao qual é destinada, seja este em formato de um interruptor, seja em formato de uma tomada ou em formato de uma caixa. As seguintes figuras mostram o diagrama de blocos de uma placa de potência de saída, de entrada e de sensores.

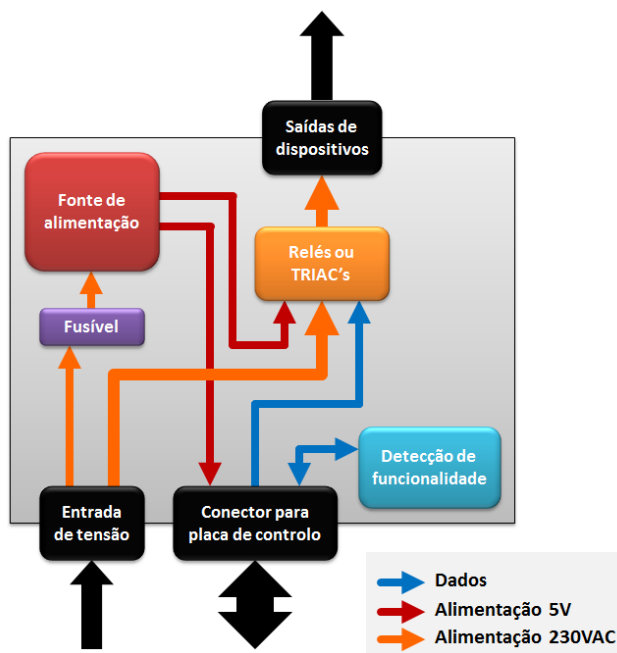


Figura 80: Diagrama de blocos da placa de potência de saída

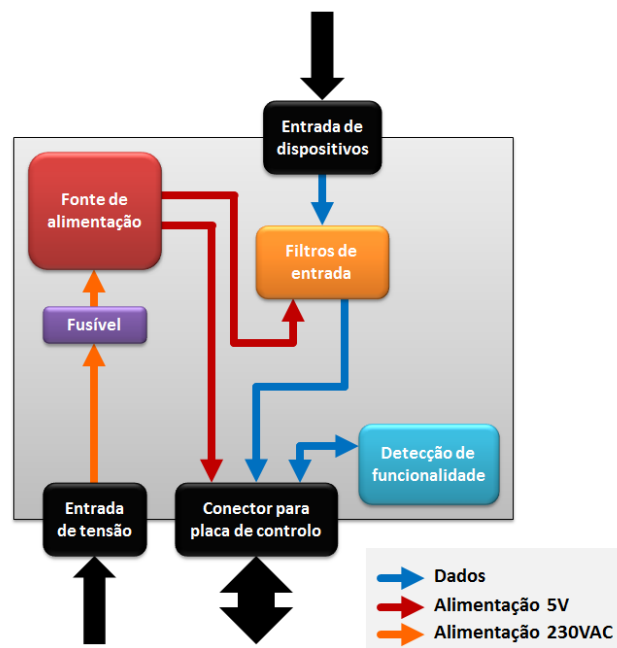


Figura 81: Diagrama de blocos da placa de potência de entrada

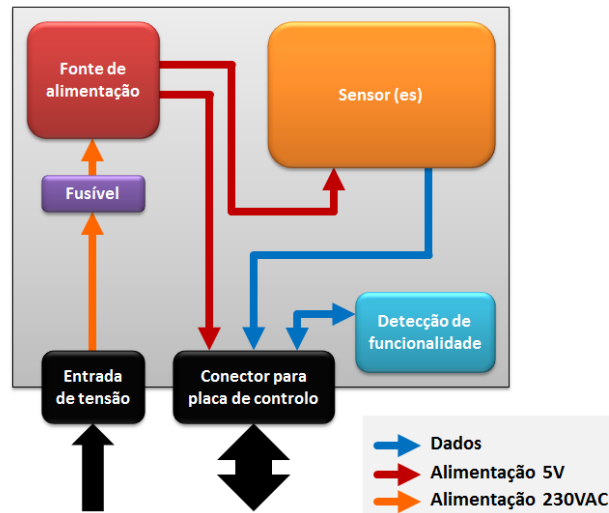


Figura 82: Diagrama de blocos da placa de potência de sensores

Dado que cada uma das placas depende da funcionalidade, o posicionamento dos componentes variará. Por essa razão, não é possível mostrar um esboço de todas as derivações, sendo apenas mostrado um traçado relativo à placa de potência de saída com relés. Pode observar-se a posição onde a placa de controlo deverá ser colocada.

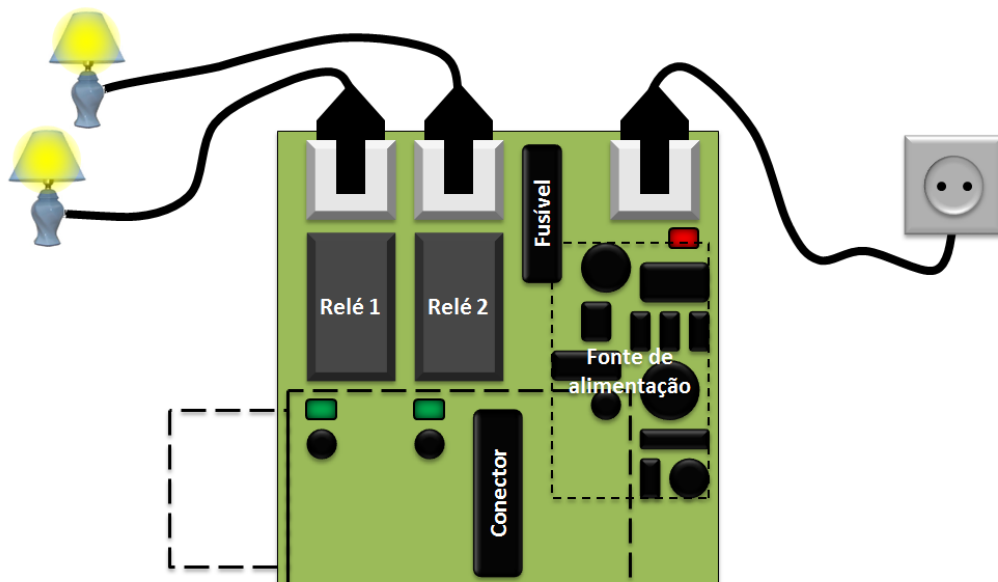


Figura 83: Simulação da organização possível para a placa de potência de saída

Detecção automática da funcionalidade

Ambas as placas, de controlo e de potência, estão equipadas com um sistema que permite identificar automaticamente qual a funcionalidade a que o módulo se destina. Esta funcionalidade refere-se ao campo *Group* existente nos campos de endereçamento do dispositivo (consultar o capítulo 3.6.3). Desta forma, é possível que o módulo processe apenas a parte de código responsável à operação necessária: sensor; interruptor; tomada.

Como o *software* do B-Live Wireless já inclui as bibliotecas referentes a todos os dispositivos, esta técnica é facilmente implementada através da utilização de uma entrada analógica que faz a interface entre as duas placas. Este pino encontra-se ligado a um divisor resistivo existente na placa de potência e constituído por duas ou mais resistências em série e/ou em paralelo, criando uma diferença de potencial que varia conforme a funcionalidade. Este potencial é medido pela placa de controlo sendo assim possível determinar o grupo. A Figura 84 ilustra o princípio de operação desta técnica.

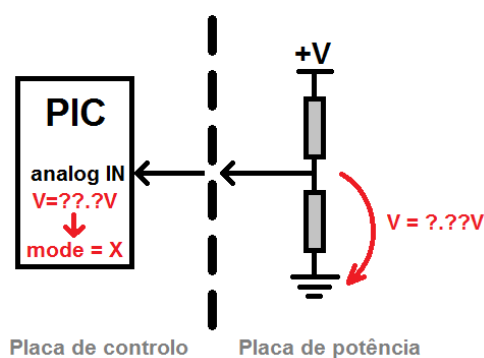


Figura 84: Técnica utilizada para detecção automática de funcionalidade

Sinais disponíveis no conector da placa de controlo

O conector existente na placa de controlo dispõe de pinos que são ligados às placas de potência ou a outra placa, seja ela de testes ou de programação. Estes pinos incluem ligações de alimentação, comunicação série, entradas e saídas e terminais de programação, entre outros. A listagem dos pinos pode ser consultada na Tabela 39, disponível no Anexo 7.

Excepções relativas à utilização de duas placas

Apesar de existirem dois tipos diferentes de placas, há casos em que tal separação é desnecessária, como é o caso de comandos remotos ou interruptores de parede, nos quais o nível de integração tem de ser mais elevado do que o normal. Nessas situações, opta-se por produzir uma placa única que contenha todos os dispositivos integrados.

5. Conclusão e trabalho futuro

O envelhecimento da população tem resultado numa pressão cada vez maior sobre as organizações que suportam serviços de saúde e de acção social. Esta dissertação enquadra-se no projecto Living Usability Lab que pretende permitir à população idosa permanecer no conforto das suas habitações reduzindo, assim, a dependência de serviços de saúde formais. Neste âmbito, foi especificado um protocolo de comunicações capaz de suportar sensores e actuadores para monitorizar o ambiente, a ocorrência de eventos, a localização e movimento de pessoas e a sua saúde. O sistema B-Live é um projecto de domótica habitacional dirigido a pessoas com limitações funcionais e foi usado como o ponto de partida para o desenvolvimento do protocolo *wireless* relatado nesta tese. O protocolo desenvolvido diferencia-se pela capacidade de suportar fluxos de tráfego com requisitos de qualidade de serviço diferenciados, mesmo em ambientes abertos, nos quais outras tecnologias de contenção podem competir pelo meio.

Esta dissertação especifica o protocolo B-Live Wireless com ênfase na integração dos diferentes tipos de serviços sob uma plataforma comum de comunicações sem-fios locais. Adicionalmente, apresenta a arquitectura de *software* desenvolvida e o estudo da arquitectura de *hardware* de suporte ao sistema B-Live Wireless. Embora não tenha conseguido concluir o desenvolvimento de todo o *software* e *hardware* necessário para a construção de um protótipo funcional, foi possível implementar e testar as bibliotecas de suporte ao protocolo e as bibliotecas de nível inferior bem como propor uma arquitectura de *hardware* com suporte para sensores e actuadores heterogéneos, usando um módulo “inteligente” comum a todos os dispositivos. As limitações de tempo impostas pelo calendário da dissertação e a complexidade de desenvolvimento de um protocolo com todas as funcionalidades requeridas impossibilitaram a apresentação, nesta fase, de um demonstrador funcional bem como a obtenção de resultados experimentais para avaliar o seu desempenho. Contudo, como se verá em seguida, este trabalho tem continuidade no projecto LUL, onde se deverá concluir o respectivo desenvolvimento.

Trabalho futuro

Como trabalho futuro, dever-se-á finalizar o desenvolvimento do *software* de suporte (WFTT) e a aplicação, implementando também a plataforma de *hardware* proposta. Seguidamente, deve ser realizada a avaliação de desempenho do protocolo em suportar os serviços identificados. A integração do sistema Wireless B-Live no projecto LUL, com outras tecnologias aí desenvolvidas, é também uma tarefa importante a realizar futuramente.

Bibliografia

- [1] Coughlin J., "Technology needs of aging boomers," *Issues in Science and Technology*, vol. 16, no. 1, pp. 53-60, 1999.
- [2] Paolo Dario, Silvestro Micera, Giovanna Macrì, Jacopo Carpaneto, and M. Chiara Carrozza, "Biorobotics for Longevity," in *Proceedings of the 2005 IEEE 9th International Conference on Rehabilitation Robotics*, Chicago IL, USA, July 1, 2005.
- [3] P. Harmp, T. Taipalus, J. Knuuttila, J. Vallet, and A. Halme, "Needs and Solutions - Home Automation and Service Robots for the Elderly and Disabled," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, p. 6.
- [4] Paulo Bartolomeu, José Fonseca, Nelson Rocha, and Filipe Basto, *Communications in Medical Applications*, 2nd ed., Bodgan Wilamowski and J. David Irwin, Eds.: Industrial Electronics Handbook, 2011.
- [5] António Teixeira et al., "A New Living Lab for Usability Evaluation of ICT and New Generation Networks for Elderly@Home," in *BIOSTEC'2011 - 4th International Conference on Biomedical Engineering Systems and Technologies*, Rome, Italy, January 2011.
- [6] EnOcean. (2011, Maio) Home page. [Online]. <http://www.enocean.com/>
- [7] Insteon. (2011, Abril) Home page. [Online]. <http://www.insteon.net/>
- [8] ONE-NET. (2011, Março) Home page. [Online]. <http://www.one-net.info/index.php>
- [9] B&B electronics. (2011, Março) Everything you wanted to know about the 868MHz radio. But were afraid to ask. [Online]. http://www.bb-europe.com/tech_articles/WP35_868MHz_Radio_Modems.asp
- [10] P. Bartolomeu and J. A. Fonseca, "Channel Capture in Noisy Wireless Contention-Based Communication Environments," , France, Maio 2010, pp. 23-32.
- [11] P. Bartolomeu, J. A. Fonseca, and F. Vasques, "Implementing the Wireless FTT Protocol: A feasibility Analysis," *IEEE*, 2010.
- [12] J. Sobrinho and A. Krishnakumar, "EQuB-Ethernet quality of service using black bursts," in *Local Computer Networks 1988. LCN '98. Proceedings. 23th Anual Conference*, França, Outubro 1988, pp. 286-296.
- [13] Microchip. (2010, Junho) Home page. [Online]. <http://www.microchip.com/>
- [14] Microchip. (2010, Março) Microchip MPLAB X IDE. [Online]. http://ww1.microchip.com/downloads/mplab/X_Beta/index.html
- [15] Microchip. (2010, Fevereiro) Microchip - Wireless and Wired Solutions. [Online]. http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2664¶m=en520414
- [16] Microchip. (2011, Fevereiro) Microchip - Application Notes. [Online]. http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en536181
- [17] Micro I/O. (2011, Março) Micro I/O - Serviços de Electrónica, Lda. [Online]. <http://www.microio.pt/>

Anexos

Anexo 1: Estrutura de um pacote MAC IEEE802.15.4

A Figura 85 ilustra um pacote definido pelo *standard* IEEE802.15.4. O protocolo B-Live Wireless utiliza uma estrutura de pacote idêntica a esta. Na figura, sobre cada um dos campos constituintes, encontra-se o tamanho deste, em *bytes*.



Figura 85: Formato do pacote do *standard* IEEE802.15.4

Como é observado, o pacote é composto por um campo com as suas características, que pode ser observado em pormenor na Figura 86. Seguidamente um campo constituído pelo número único identificativo do pacote, quatro campos relativos ao endereçamento do dispositivo, dois campos relativos à segurança dos dados, o campo de dados do utilizador e, finalmente, um campo para verificar a sequência do pacote.



Figura 86: Definições associadas ao campo Frame Control da Figura 85

Anexo 2: Subgrupos associados ao campo *SRS/DST Room*

A seguinte tabela expõe todas as divisões existentes que podem ser seleccionadas através dos campos: *SRS Room* (verde) para endereçamento de origem, e *DST Room* (vermelho) para endereçamento de destino.

Tabela 18: Divisões disponíveis para atribuir ao campo *SRC/DST Room*

SRC Room Divisão do dispositivo de origem		DST Room Divisão do dispositivo de destino
Valor (dec)	Valor (bin)	Descrição
0	00000 000	Dispositivo sem configuração
1 a 7	00000 001 a 00000 111	Reservado
8 a 47	00001 000 a 00101 111	Quarto 1 a 40
48 a 87	00110 000 a 01010 111	Casa de banho 1 a 40
88 a 95	01011 000 a 01011 111	Cozinha 1 a 8
96 a 103	01100 000 a 01100 111	Despensa 1 a 8
104 a 111	01101 000 a 01101 111	Sala 1 a 8
112 a 119	01110 000 a 01110 111	Corredor 1 a 8
120 a 127	01111 000 a 01111 111	Escadas 1 a 8
128 a 135	10000 000 a 10000 111	Varanda 1 a 8
136 a 146	10001 000 a 10001 111	Garagem 1 a 8
144 a 151	10010 000 a 10010 111	Cave 1 a 8
152 a 159	10011 000 a 10011 111	Sótão 1 a 8
160 a 167	10100 000 a 10100 111	Armazém 1 a 8
168 a 175	10101 000 a 10101 111	Exterior 1 a 8
176 a 183	10110 000 a 10110 111	Jardim 1 a 8
184 a 191	10111 000 a 10111 111	Outros 1 a 8
192 a 247	11000 000 a 11110 111	Livre
248 a 252	11111 000 a 00000 100	Reservado
253	11111 101	Dispositivo móvel
254	11111 110	Dispositivo de controlo de sector (<i>Sector</i>)
255	11111 111	Reservado

Anexo 3: Subgrupos associados ao campo *SRS/DST Group*

As seguintes tabelas expõem todos os subgrupos existentes que podem ser seleccionados através dos campos: *SRS Group* (verde), para endereçamento de origem, e *DST Group* (vermelho) para endereçamento de destino.

Tabela 19: Grupos disponíveis para atribuir ao campo *SRC/DST Group*

SRC Group Grupo de dispositivos de origem		DST Group Grupo de dispositivos de destino
Valor (dec)	Valor (bin)	Descrição
0	00000 000	Dispositivo sem configuração
1 a 7	00000 001 a 00000 111	Reservado
8 a 15	00001 000 a 00001 111	Iluminação
16 a 23	00010 000 a 00010 111	Portas / Janelas
24 a 31	00011 000 a 00011 111	Estores
32 a 39	00100 000 a 00100 111	Tomadas de potência
40 a 47	00101 000 a 00101 111	Tomadas comuns
48 a 55	00110 000 a 00110 111	Autoclismo
56 a 63	00111 000 a 00111 111	Torneiras
64 a 71	01000 000 a 01000 111	Menu vocal
72 a 79	01001 000 a 01001 111	Intercomunicadores
80 a 87	01010 000 a 01010 111	Comando multimédia
88 a 95	01011 000 a 01011 111	Dispensadores
96 a 103	01100 000 a 01100 111	Ventilação / Aquecimento
104 a 111	01101 000 a 01101 111	Interruptores
112 a 119	01110 000 a 01110 111	Dispositivos móveis
120 a 127	01111 000 a 01111 111	Elevador
128 a 142	10000 000 a 10001 111	Sensores
143 a 247	10010 000 a 11110 111	Livre
248 a 253	11111 000 a 11111 101	Reservado
254	11111 110	Dispositivo de controlo de divisão (Room)
255	11111 111	Reservado

O primeiro elemento de cada uma das tabelas seguintes corresponde ao nome principal do subgrupo exibido na Tabela 19. Encontram-se representados os campos relativos a origem (verde) e a destino (vermelho).

Tabela 20: Subgrupo de iluminação do campo *SRC/DST Group*

Iluminação	Iluminação
Valor (dec)	Descrição
8	Iluminação geral
9	Iluminação de parede
10	Iluminação de tecto
11	Iluminação decorativa
12	Iluminação exterior
13	Livre
14	Livre
15	Iluminação de emergência

Tabela 21: Subgrupo de portas e janelas do campo *SRC/DST Group*

Portas e janelas	Portas e janelas
Valor (dec)	Descrição
16	Portas e janelas geral
17	Porta exterior
18	Porta interior
19	Janela
20	Portão
21	Livre
22	Livre
23	Porta de emergência

Tabela 22: Subgrupo de estores do campo *SRC/DST Group*

Subgrupo de estores	Subgrupo de estores
Valor (dec)	Descrição
24	Estores geral
25	Estores de janela
26	Estores da porta
27	Livre
28	Livre
29	Livre
30	Livre
31	Livre

Tabela 23: Subgrupo de tomadas de potência do campo *SRC/DST Group*

Tomadas de potência	Tomadas de potência
Valor (dec)	Descrição
32	Tomada de potência geral
33	Fogão
34	Forno
35	Máquina de lavar louça
36	Máquina de lavar roupa
37	Máquina de secar roupa
38	Livre
39	Livre

Tabela 24: Subgrupo de tomadas comuns do campo *SRC/DST Group*

Tomadas comuns	Tomadas comuns
Valor (dec)	Descrição
40	Tomada comum geral
41	Frigorífico
42	Microondas
43	Torradeira
44	Máquina de café
45	Grelhador
46	Livre
47	Livre

Tabela 25: Subgrupo de autoclismo do campo *SRC/DST Group*

Autoclismo	Autoclismo
Valor (dec)	Descrição
48	Autoclismo geral
49	Controlo de descarga
50	Controlo da tampa
51	Livre
52	Livre
53	Livre
54	Livre
55	Livre

Tabela 26: Subgrupo de torneiras do campo *SRC/DST Group*

Torneiras	Torneiras
Valor (dec)	Descrição
56	Torneira geral
57	Torneira de lava-louça
58	Torneira de lavatório
59	Torneira de bidé
60	Torneira de banheira
61	Torneira de rega
62	Livre
63	Aspersor de emergência

Tabela 27: Subgrupo de menu vocal do campo *SRC/DST Group*

Menu vocal	Menu vocal
Valor (dec)	Descrição
64	Menu vocal geral
65	Livre
66	Livre
67	Livre
68	Livre
69	Livre
70	Livre
71	Livre

Tabela 28: Subgrupo de intercomunicador do campo *SRC/DST Group*

Intercomunicador	Intercomunicador
Valor (dec)	Descrição
72	Intercomunicador geral
73	Intercomunicador porteiro
74	Livre
75	Livre
76	Livre
77	Livre
78	Livre
79	Livre

Tabela 29: Subgrupo de multimédia do campo *SRC/DST Group*

Multimédia	Multimédia
Valor (dec)	Descrição
80	Controlo multimédia geral
81	Controlo de TV
82	Controlo de DVD/Vídeo
83	Controlo de Áudio
84	Controlo de telefone
85	Livre
86	Livre
87	Livre

Tabela 30: Subgrupo de dispensadores do campo *SRC/DST Group*

Dispensadores	Dispensadores
Valor (dec)	Descrição
88	Dispensador geral
89	Dispensador de alimentos
90	Dispensador de bebidas
91	Dispensador de papel
92	Livre
93	Livre
94	Livre
95	Livre

Tabela 31: Subgrupo de ventilação e aquecimento do campo *SRC/DST Group*

Ventilação / aquecimento	Ventilação / aquecimento
Valor (dec)	Descrição
96	Ventilação e aquecimento geral
97	Ar condicionado
98	Ventilador
99	Exaustor de fumos
100	Aquecimento eléctrico
101	Livre
102	Livre
103	Livre

Tabela 32: Subgrupo de interruptores do campo *SRC/DST Group*

Interruptores	Interruptores
Valor (dec)	Descrição
104	Interruptor geral
105	Interruptor simples
106	Interruptor duplo
107	<i>Dimmer</i> simples
108	<i>Dimmer</i> duplo
109	Livre
110	Livre
111	Livre

Tabela 33: Subgrupo de dispositivos móveis do campo *SRC/DST Group*

Dispositivos móveis	Dispositivos móveis
Valor (dec)	Descrição
112	Dispositivo móvel geral
113	SmartRemote geral
114	Localizador tag
115	Monitor de sinais vitais
116	Livre
117	Livre
118	Livre
119	Livre

Tabela 34: Subgrupo de elevador do campo *SRC/DST Group*

Elevador	Elevador
Valor (dec)	Descrição
120	Elevador geral
121	Elevador de pessoas
122	Elevador monta-cargas
123	Grua elevatória
124	Livre
125	Livre
126	Livre
127	Livre

Tabela 35: Subgrupo de sensores do campo *SRC/DST Group*

Sensores	Sensores
Valor (dec)	Descrição
128	Sensor geral 1
129	Sensor de fumo
130	Sensor de gás
131	Sensor de inundação
132	Sensor de iluminação
133	Sensor de corrente
134	Sensor de tensão
135	Sensor de queda
136	Sensor geral 2
137	Sensor de movimento
138	Sensor de pressão
139	Sensor magnético
140	Livre
141	Livre
142	Livre
143	Livre

NOTA: Os campos de *DST/SRC Room* com valores desde 144 até 247 (13 subgrupos) encontram-se livres para futuras implementações.

Anexo 4: Valores atribuídos a campos da trama B-Live Wireless

Tabela 36: Valores associados ao campo *OUT* da trama de ordem de operação normal

OUT <i>n</i> – Estado da saída <i>n</i>	
Valor (hex)	Descrição
0x00	Sem qualquer alteração
0x01	Desliga a saída
0x02 a 0xFD	Varia a intensidade da saída (opcional)
0xFE	Liga a saída
0xFF	Inverte o estado da saída

Tabela 37: Descrição das causas de erro da trama de diagnóstico de erro

Cause byte 1/2 – Causa de erro do dispositivo		
ID	byte 1 (bin) – byte 2 (bin)	Descrição
0	00000000 – 00000000	Indeterminado
1	XXXXXXXX – XXXXXXXX1	Problemas com a saída 1
2	XXXXXXXX – XXXXXXXX1X	Problemas com a saída 2
3	XXXXXXXX – XXXXXXXX1XX	Problemas com a saída 3
4	XXXXXXXX – XXXX1XXX	Problemas com a entrada 1
5	XXXXXXXX – XXX1XXXX	Problemas com a entrada 2
6	XXXXXXXX – XX1XXXXX	Problemas com a entrada 3
7	XXXXXXXX – X1XXXXXX	Livre
8	XXXXXXXX – 1XXXXXXX	Livre
9	XXXXXXXX1 – XXXXXXXX	Sobreaquecimento
10	XXXXXX1X – XXXXXXXX	Tensão demasiadamente elevada
11	XXXXXX1X – XXXXXXXX	Corrente demasiadamente elevada
12	XXXX1XXX – XXXXXXXX	Livre
13	XXX1XXXX – XXXXXXXX	Livre
14	XX1XXXXX – XXXXXXXX	Livre
15	X1XXXXXX – XXXXXXXX	Livre
16	1XXXXXXX – XXXXXXXX	Comunicação mal sucedida

Tabela 38: Valores associados ao campo *TX Power* da trama de configuração

<i>TX Power</i> – Selecção da potência de operação		
Valor (dec)	Valor (hex)	Descrição
0 a 127	00000000 - 01111111	Sem qualquer alteração
-	100 XX 000	Potência em pequena escala: 0.0dB
-	100 XX 001	Potência em pequena escala: 0.5dB
-	100 XX 010	Potência em pequena escala: 1.2dB
-	100 XX 011	Potência em pequena escala: 1.9dB
-	100 XX 100	Potência em pequena escala: 2.8dB
-	100 XX 101	Potência em pequena escala: 3.7dB
-	100 XX 110	Potência em pequena escala: 4.9dB
-	100 XX 111	Potência em pequena escala: 6.3dB
-	100 00 XXX	Potência em grande escala: 0dB
-	100 01 XXX	Potência em grande escala: 10dB
-	100 10 XXX	Potência em grande escala: 20dB
-	100 11 XXX	Potência em grande escala: 30dB
160 a 255	101 00 000 – 111 11 111	Sem qualquer alteração

Anexo 5: Tutorial de configuração do ambiente de desenvolvimento

Ponto I – Instalação do *IDE MPLAB X* em ambiente *Linux*

Enquadramento

Este documento destina-se a elucidar os procedimentos relativos à instalação e configuração do ambiente de desenvolvimento MPLAB X [1], assim como dos compiladores de C em ambiente *Linux*. Foi utilizado o sistema *Linux Ubuntu* 10.10, e a versão do MPLAB X *beta* 5.0.

Guia passo a passo

1) Download do software necessário

Efectue o *download* do MPLAB X e dos compiladores necessários para a versão *Linux* através do endereço seguinte [2]:

http://ww1.microchip.com/downloads/mplab/X_Beta/index.html



Figura 87: Elementos necessários à instalação do MPLAB X

Os ficheiros são do tipo *.bin. Guarde-os, por exemplo, na pasta pessoal.

2) Instalar o JAVA runtime environment

Neste momento, é necessário ter o *JAVA JRE* instalado. Habitualmente as máquinas equipadas com uma versão actualizada do *Linux* já contém o "*openJDK*", no entanto o *MPLAB X* necessita da versão da *Sun*. Consequentemente é necessário desinstalar o *openJDK* [3].

Abra o *Synaptic Package Manager* e pesquise por "*openjdk*":

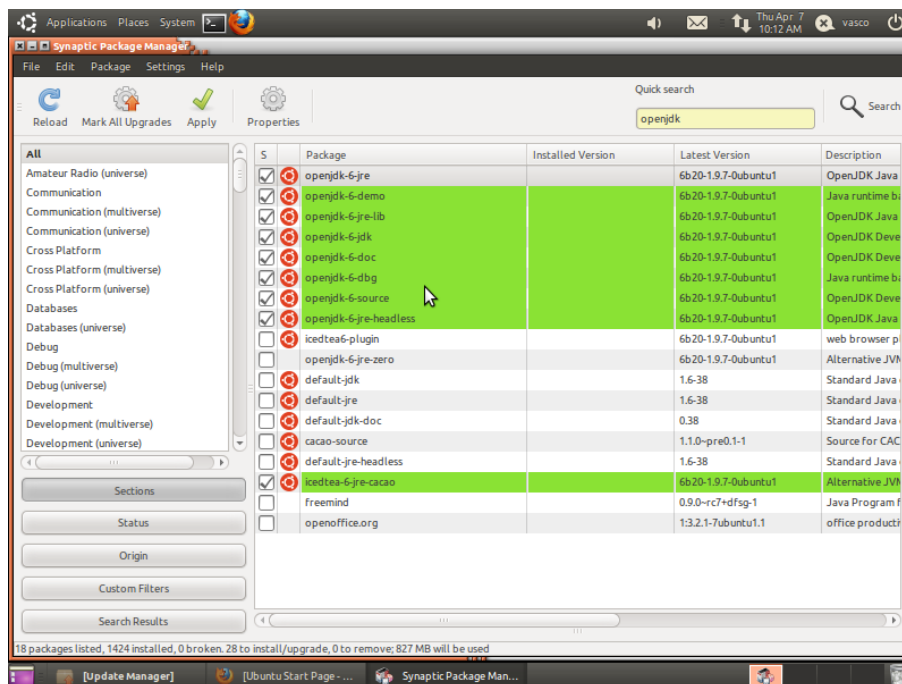


Figura 88: Pacotes do *openjdk* necessários para o MPLAB X

Marque para desinstalação e clique em *Apply*.

Depois seleccione "*Repositories*" no menu "*Settings*", na janela seleccione a *tab* "*Other software*". Seleccione os repositórios abaixo indicados:

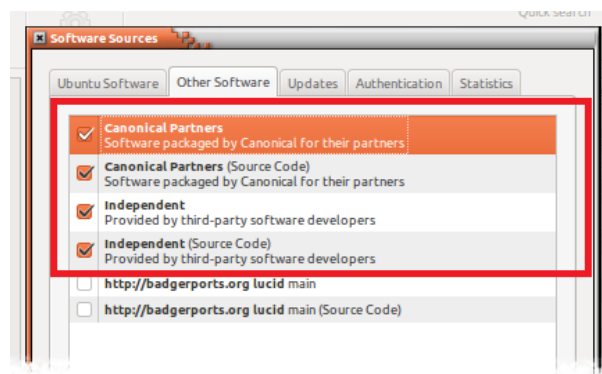


Figura 89: Repositórios necessários para o MPLAB X

Volte à janela principal e clique em "*Reload*". Pesquise na lista de pacotes o texto "*sun javd*":

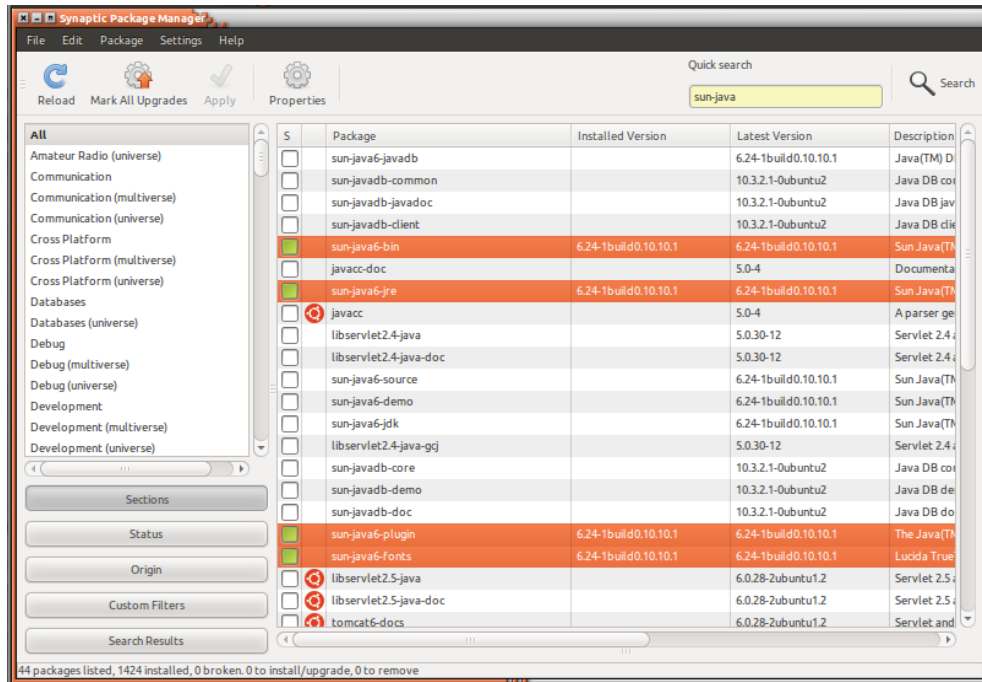


Figura 90: Instalação de pacotes necessários para o MPLAB X

Localize os pacotes mostrados em cima e marque-os para instalação. Aplique as alterações e feche a janela.

3) Instalar o MPLABX e os compiladores

Abra uma consola e aponte o directório desta para a pasta onde colocou os ficheiros de instalação do MPLAB X e dos compiladores.

Digite na consola o comando em baixo para os ficheiros que fez *download* (neste caso o MPLAB X, C18 e C30). Este comando vai atribuir perdições de execução para que possa instalar o software:

```
chmod +x <nome do ficheiro.bin>
```

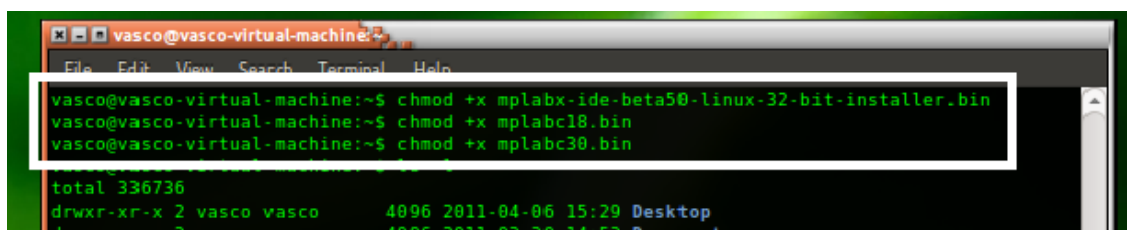


Figura 91: Comandos de consola necessários para executar a instalação do MPLAB X

Na mesma consola, execute o instalador do MPLAB X. Para este efeito escreva:

```
sudo ./<nome do ficheiro.bin>
```

Insira a *password* e o processo de instalação começa:

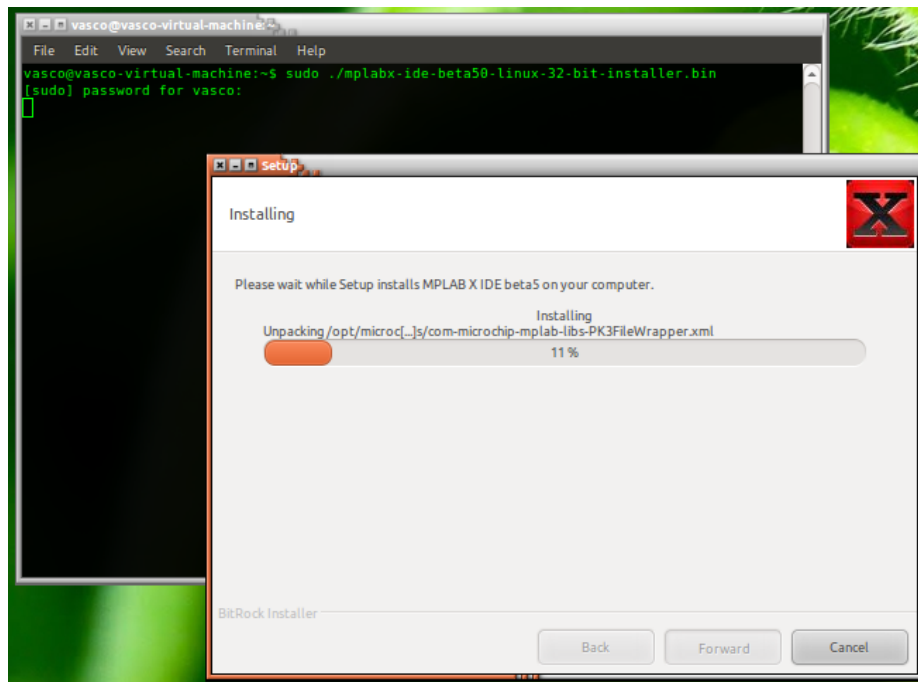


Figura 92: Instalação do IDE MPLAB X

Aguarde que a instalação do MPLAB X termine. Após ter terminado, instale os compiladores de forma semelhante através do comando descrito no início da página:

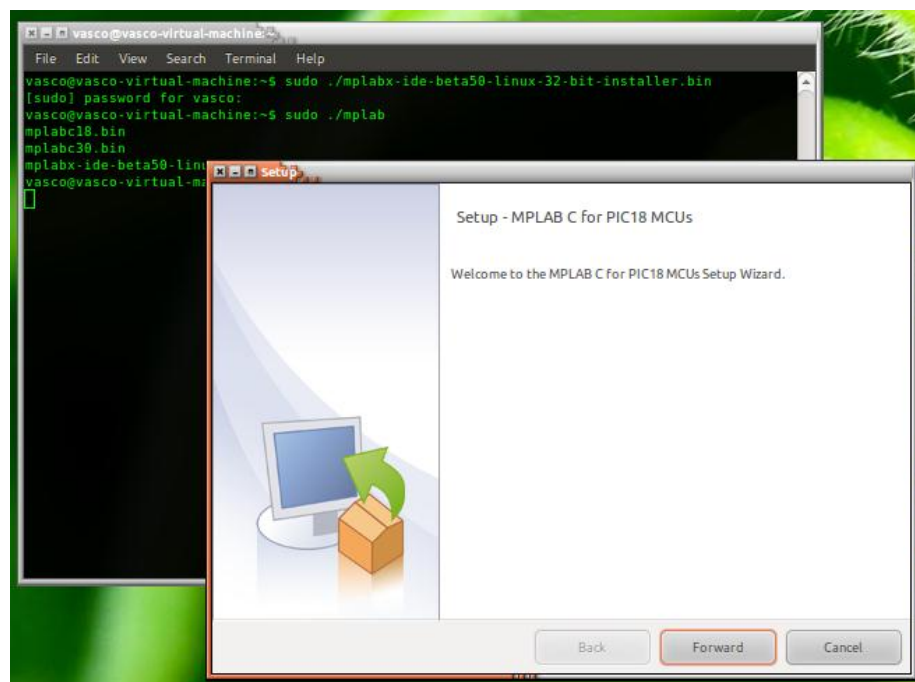


Figura 93: Instalação dos compiladores do IDE MPLAB X

Reinicie o sistema. O MPLAB X já está instalado e pronto a ser utilizado.

Conclusões e notas importantes

O MPLAB X funciona em *Linux* exactamente da mesma forma que em *Windows*. Apenas deverá que ter em atenção os seguintes pontos:

PROBLEMA: Na compilação poderão existir erros em *includes*, por exemplo, em que o compilador não consegue encontrar os ficheiros.

CAUSA: O sistema de ficheiros do *Linux* é *case sensitive*.

SOLUÇÃO: Porque em *Linux* "mcp.h" e "MCP.h" não são o mesmo ficheiro. Terá de alterar todos os nomes de forma a ficarem uniformes.

PROBLEMA: Podem surgir erros com pontuação, como ".".

CAUSA: Possivelmente devido ao *encoding* de texto.

SOLUÇÃO: Basta apagar e escrever de novo o símbolo.

PROBLEMA: Podem surgir erros de compilação em *includes*, por exemplo, devido a barras '/'.

CAUSA: As barras '/' tipicamente utilizadas em *Windows* para definir caminhos, são agora '\' em *Linux*.

SOLUÇÃO: Substitua as barras '/' por barras '\'.

Tome atenção ao facto de portabilidade de projectos entre sistemas *Windows* ↔ *Linux*.

Ponto II – Definir os *configuration bits* no MPLAB X

Enquadramento

Este documento destina-se a expor os procedimentos relativos à configuração dos *configuration bits*, ou *fuses*, dos PIC's para que o código do utilizador possa ser programado directamente por *ICSP*, sem recorrer a um *bootloader*. Para esta exemplificação foi utilizado o compilador C18.

Como a opção gráfica desapareceu da versão MPLAB para a MPLAB X, torna-se agora necessário introduzir as configurações através de texto no próprio código.

Guia passo a passo

1) Obter as opções de configuração

Após ter instalado o compilador C18, navegue até à pasta seguinte e abra o ficheiro "*hlpPIC18ConfigSet.chm*":

C:\MCC18\doc

Localize a página referente ao PIC que está a utilizar. Neste caso foi utilizado um PIC18F26K20.

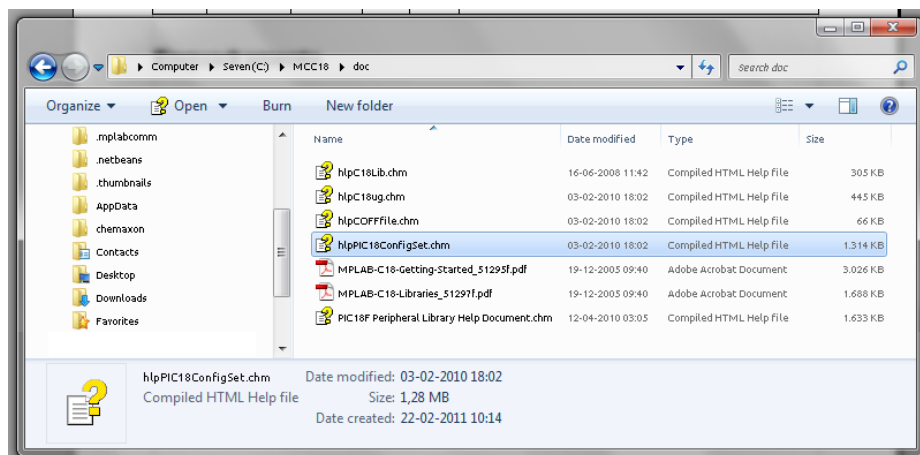


Figura 94: Pasta dos ficheiros de documentação do compilador C18

Agora basta apenas seleccionar as opções desejadas da lista e copiá-las para o código aberto no MPLAB X, para o ficheiro "*main.c*" imediatamente no início do programa. Deve ser utilizada a formatação seguinte:

```
#pragma config OPCA01 = VALOR1
#pragma config OPCA02 = VALOR2
#pragma config OPCA0n = VALORn
```

As imagens seguintes mostram um exemplo de configurações retiradas do documento, inseridas no projecto do MPLAB X:

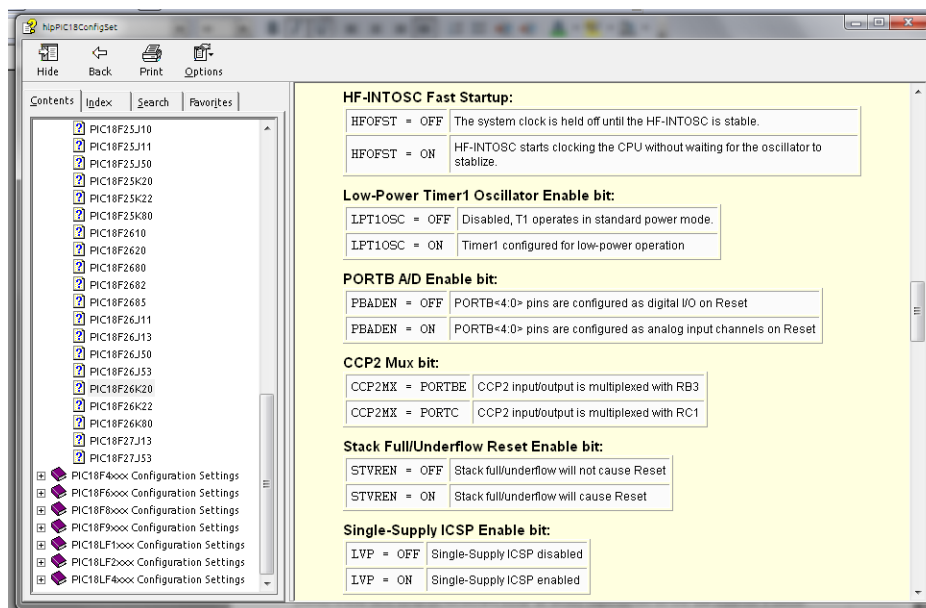
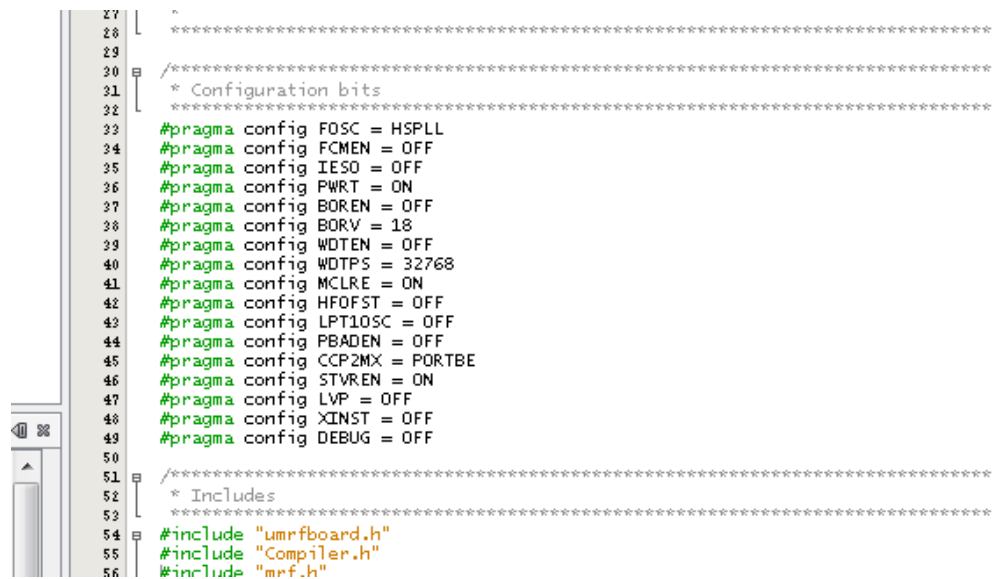


Figura 95: Documentação dos *configuration bits* para o PIC utilizado

As configurações desejadas são:

FOSC = HSPLL
 FC MEN = OFF
 IESO = OFF
 PWRT = ON
 BOREN = OFF
 BORV = 18
 WDTEN = OFF
 WDTPS = 32768
 MCLRE = ON
 HFOFST = OFF
 LPT1OSC = OFF
 PBA DEN = OFF
 CCP2MX = PORTBE
 STVREN = ON
 LVP = OFF
 XINST = OFF
 DEBUG = OFF

As configurações anteriores inseridas no projecto do MPLABX:

A screenshot of the MPLABX code editor interface. The editor shows a C source file with configuration bits and includes. The line numbers 27 through 56 are visible on the left margin. The code is as follows:

```
27  /*  
28  * Configuration bits  
29  */  
30  /*  
31  * Configuration bits  
32  */  
33  #pragma config FOSC = HSPLL  
34  #pragma config FCMEN = OFF  
35  #pragma config IESO = OFF  
36  #pragma config PWRT = ON  
37  #pragma config BOREN = OFF  
38  #pragma config BORV = 18  
39  #pragma config WDTPS = 32768  
40  #pragma config MCLRE = ON  
41  #pragma config HFOFST = OFF  
42  #pragma config LPT1OSC = OFF  
43  #pragma config PBAEN = OFF  
44  #pragma config CCP2MX = PORTBE  
45  #pragma config STVREN = ON  
46  #pragma config LVP = OFF  
47  #pragma config XINST = OFF  
48  #pragma config DEBUG = OFF  
49  
50  
51  /*  
52  * Includes  
53  */  
54  #include "umrfboard.h"  
55  #include "Compiler.h"  
56  #include "mrf.h"
```

Figura 96: Configuration bits introduzidos no código

Desta forma, torna-se possível que o projecto funcione na placa de destino, independentemente da existência ou não de um *bootloader* no PIC.

Ponto III – Instalação e utilização do ICD2 ou ICD3 no MPLAB X

Enquadramento

Este documento destina-se a explicar os procedimentos relativamente à instalação e utilização do programador e *debugger* ICD 2 ou 3 da Microchip num computador com Windows e com os softwares MPLAB v8.x ou superior ou MPLABX beta 5.0 ou superior.

Guia passo a passo

1) Instalação do software necessário

Para instalar o ICD2 e o ICD3 no MPLAB X necessita da versão normal do MPLAB instalada, devido aos *drivers*. Caso não disponha do MPLAB utilizando o *link* em baixo, faça *download* da versão mais recente do site da *Microchip*®.

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

2) Ligação do ICD 2 ou 3 ao computador

Ligue o ICD 2 ou o ICD 3 ao computador através da porta correspondente. Neste momento, o computador irá procurar pelas *drivers*. Caso estas não sejam encontradas, é necessário procura-las manualmente e, nesse caso, elas encontram-se na seguinte pasta:

C:\Program Files\Microchip\MPLAB IDE\ICD2\Drivers

C:\Program Files\Microchip\MPLAB IDE\ICD3\Drivers

Selecione a pasta correcta e instale o *driver* correspondente.

Neste momento o ICD2/3 já está configurado e pronto a ser utilizado no MPLAB. No entanto, é necessário configurar o sistema para que o ICD2/3 possa ser utilizado no MPLABX.

3) Configuração do ICD 2 ou 3 no MPLABX

O ICD 2/ 3 quando se encontra ligado ao sistema, comunica com o MPLAB através de um *driver* proprietário da *Microchip*®, o qual vem incluído com a instalação deste IDE.

No caso do MPLAB X, o ICD utiliza uma *driver* genérica de *Windows: WinUSB*.

Consequentemente, torna-se necessário trocar as *drivers* dependendo do IDE que estamos a utilizar.

Felizmente, o MPLABX contém uma pequena aplicação que rapidamente faz a troca dos *drivers* [4]. Esta aplicação encontra-se em:

Versões 32-bit Windows:

C:\Program Files\Microchip\MPLAB X IDE\
Switcher\32bit\MPDDSwich32.exe

Versões 64-bit Windows:

C:\Program Files\Microchip\MPLAB X IDE\
Switcher\64bit\MPDDSwich64.exe

Na aplicação podem ser seleccionados qual dos IDE's irá utilizar o ICD2/3.

(Nota: A aplicação é compatível também com o Real ICE, PM3 e PIC32 SK)

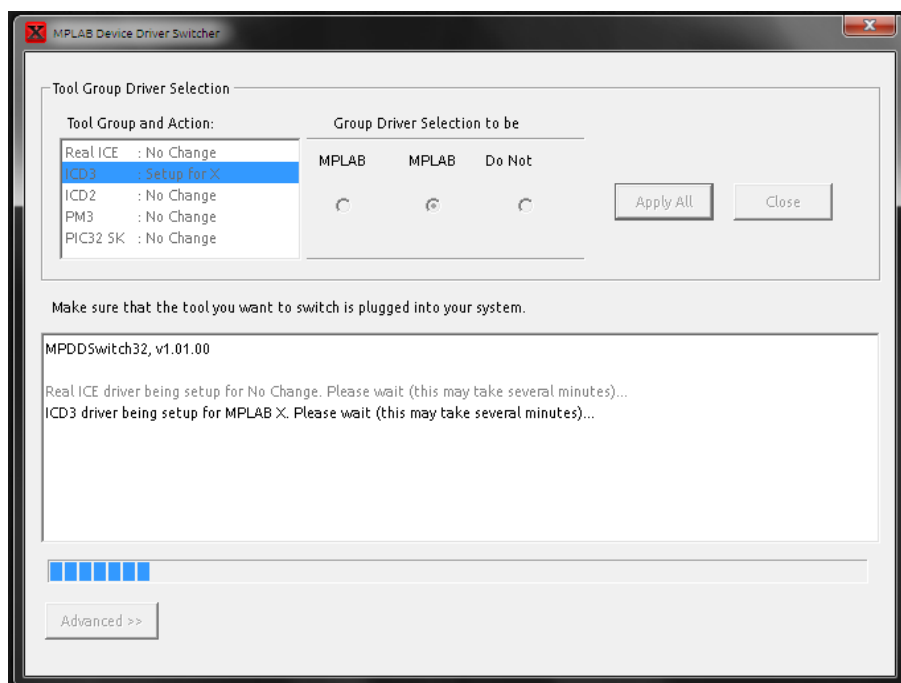


Figura 97: Aplicação para configurar os *drivers* dos programadores

Agora poderá executar o MPLA X e o ICD vai, então, ser reconhecido e ficar disponível para utilização.

4) Utilização do ICD 2 ou 3 no MPLABX

Em primeiro lugar, após ter executado o MPLAB X, abra o projecto desejado. Abra as propriedades deste da seguinte forma:

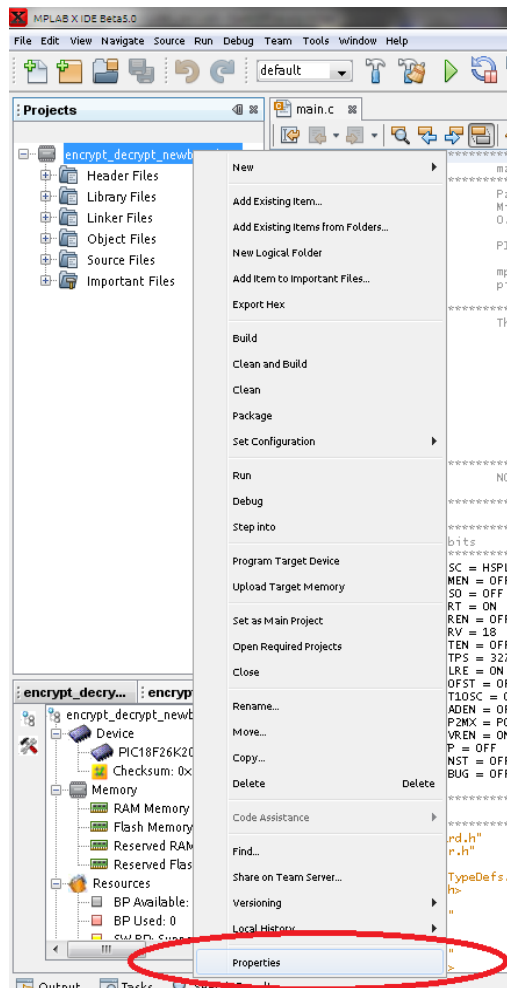


Figura 98: Aplicação para configurar os *drivers* dos programadores

Depois, irá encontrar uma janela onde, se tudo estiver correctamente instalado como descrito anteriormente, irá encontrar o ICD2/3 (3 neste caso) activo e seleccionável. Seccione-o e atribua-lhe um *friendly name* caso seja necessário. Aplique as alterações.

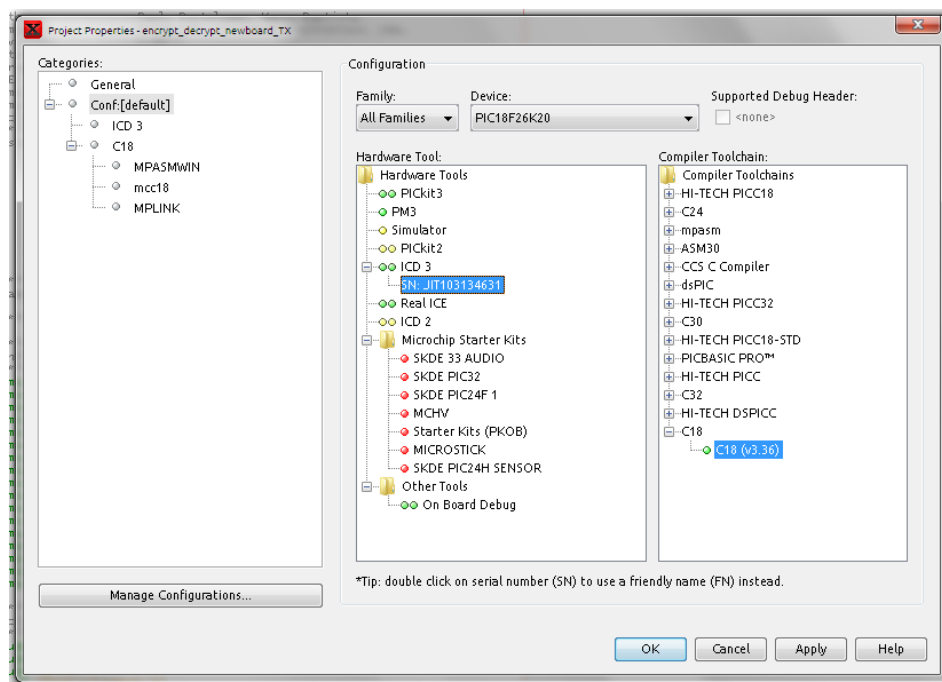


Figura 99: Janela de configuração do projecto

Caso deseje alterar as configurações relativas ao ICD, na mesma janela entre no menu referente ao dispositivo que está a utilizar.

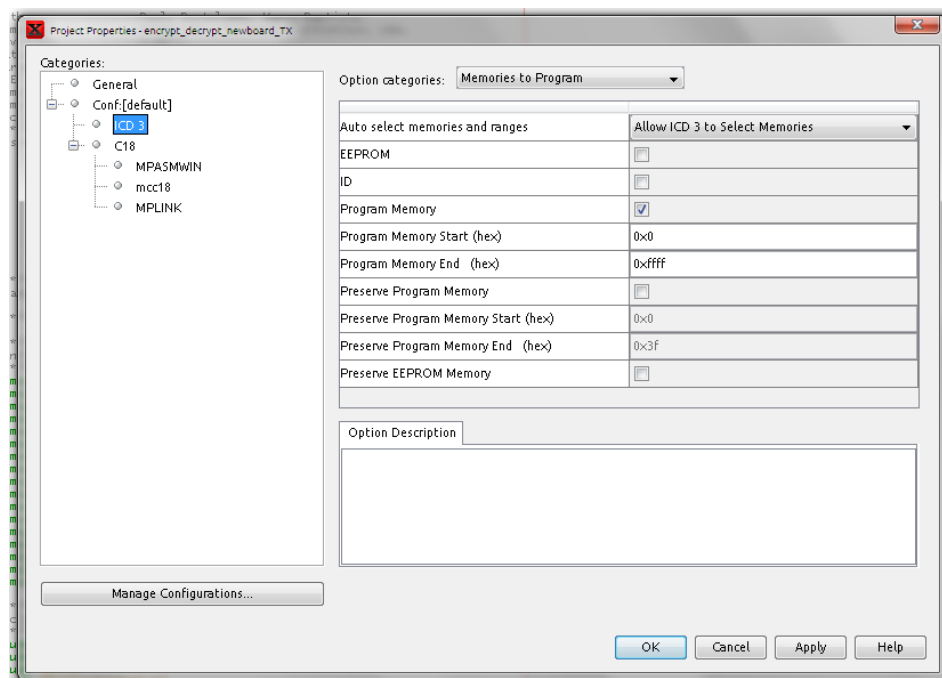


Figura 100: Janela de configuração do ICD

Faça as alterações e verificações necessárias. No final aplique as alterações e feche a janela.

Ligue agora o interface ICSP à placa que deseja programar ou efectuar *debug*. O ICD está pronto a ser utilizado neste momento. Para programar ou fazer *debug* ao seu código, utilize os botões existentes na janela na parte superior.

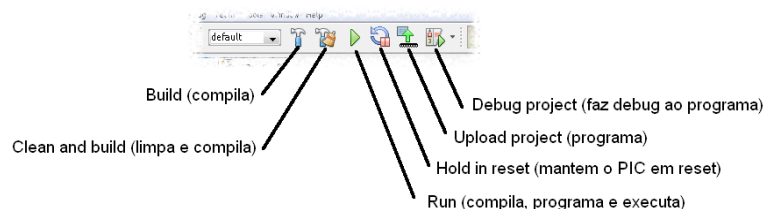


Figura 101: Botões de operação do ICD no MPLAB X

Para evitar quaisquer problemas deve retirar a ficha ICSP da placa após a programação, de forma a evitar possíveis curto-circuitos.

Ponto IV – Instalação do *bootloader ds30 Loader* em ambiente *Linux*

Enquadramento

Este documento destina-se a explicar os procedimentos relativos à instalação e configuração software ds30 Loader [5], o software de comunicação com o bootloader utilizado por grande parte dos PICs na Micro I/O.

Guia passo a passo

1) Download do software necessário

Faça download da versão mais recente do *package* do *ds30 Loader* através do *link* seguinte (faça download da versão ZIP!):

<http://mrmackey.no-ip.org/elektronik/ds30loader/downloads.php>

This release is which contains 1.0.0 and 1.0.1.0. This version is not thoroughly tested. Upgrade is only recommended if you have problems.

Latest package		15th January 2011
Zipped	ds30 Loader 110115.zip	(2MB)
Windows installation	ds30_loader_110115.exe	(2MB)
Firmware PIC16F	1.0.0	
Firmware PIC18F	2.0.4	
Firmware PIC18FJ	1.0.4	
Firmware PIC24F	1.0.0	

Figura 102: Ficheiro de instalação existente no *website*

Guarde o ficheiro por exemplo na pasta pessoal e depois extraia os ficheiros para uma outra pasta, como se mostra em baixo:

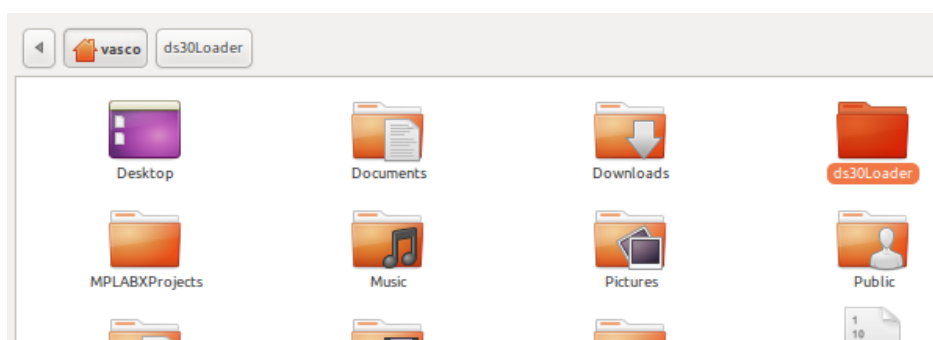


Figura 103: Pasta com os ficheiros extraídos referentes ao ds30Loader

2) Instalar o Mono

Para que consiga abrir o *software ds30 Loader* tem que instalar a ferramenta *Mono* [6], que é semelhante ao conhecido *Wine*.

Abra o "*Synaptic Package Manager*" e seleccione "*Repositories*" no menu "*Settings*", na janela seleccione a *tab* "*Other software*". Clique no botão "*Add...*" e adicione o seguinte texto [7]:

`http://badgerports.org lucid man`

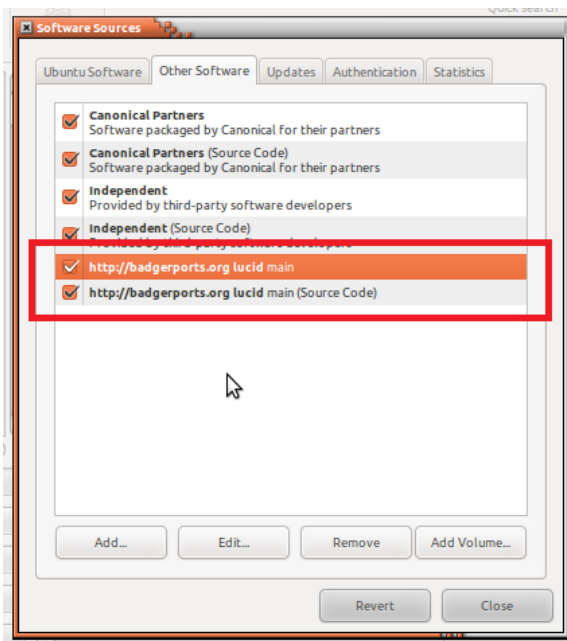


Figura 104: Repositórios necessários para a instalação do Mono

Volte à janela principal e clique em "*Reload*", depois pesquise na lista de pacotes o texto "*mono-devel*":

Selecione os pacotes "*mono-devel*" e "*mono-2.0-devel*" e marque-os para instalação. Os restantes pacotes são seleccionados automaticamente.

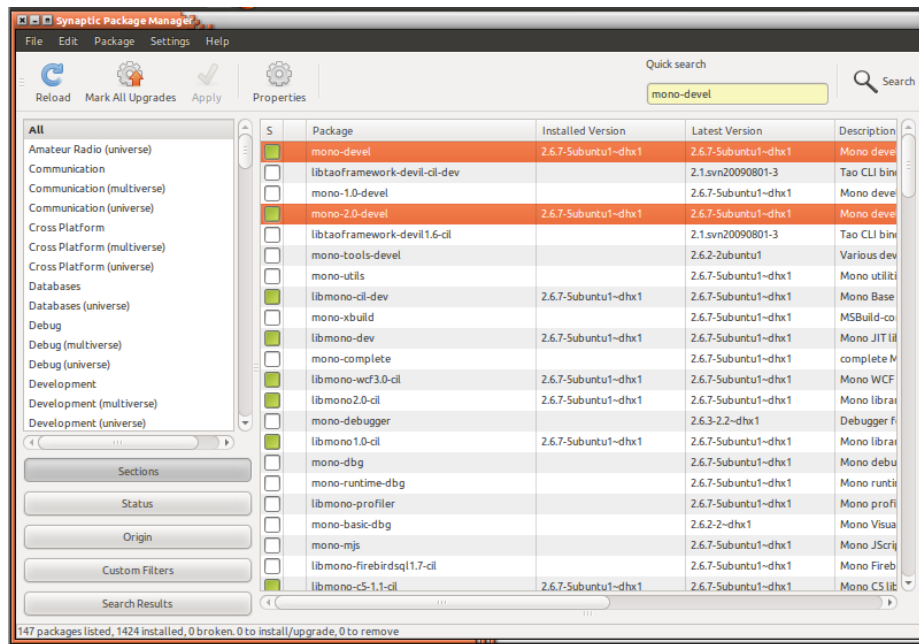


Figura 105: Pasta com os ficheiros extraídos referentes ao ds30Loader

3) Ligar a placa ao PC

Caso esteja a utilizar uma porta série física do computador não necessita de executar os comandos seguintes, pode abrir imediatamente o *software*. Apenas se destinam a verificar se o conversor USB-SERIE se encontra correctamente ligado e identificado.

Abra uma consola e vá à pasta /dev:

```
cd /dev
```

Localize as portas série do computador executando o comando:

```
ls tty*
```

Ligue agora o dispositivo através do cabo USB.

Volte a executar o comando anterior, irá notar uma diferença na lista:

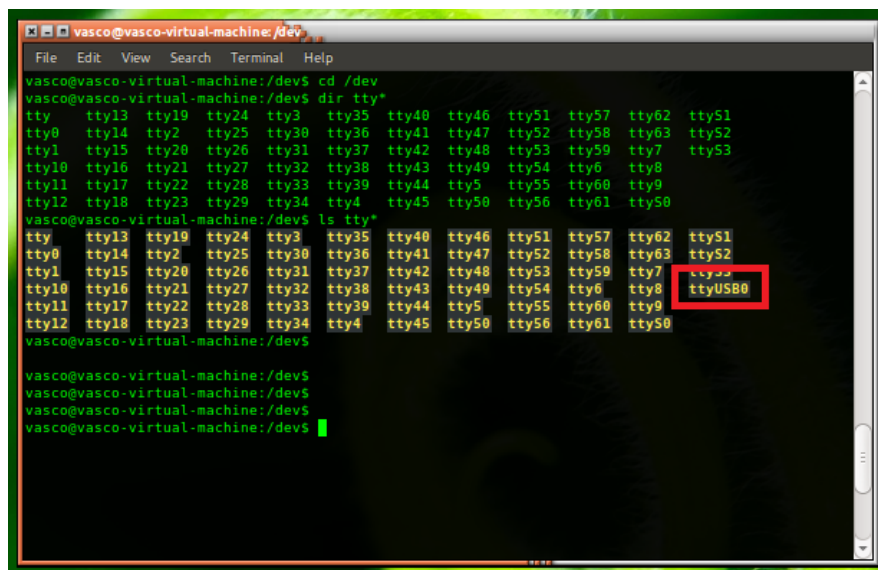


Figura 106: O dispositivo USB-SERIE é mostrado na lista de *devices*

4) Abrir o *ds30 Loader*

Abra uma nova consola e dirija-se à pasta onde se encontram os ficheiros executáveis do *ds30*. Neste caso a pasta é: "*vasco/ds30Loader/bin*".

Execute o comando seguinte para abrir o *ds30 Loader*:

```
mono ds30\ Loader\ GUI.exe
```

Neste momento o programa *ds30 Loader* deve mostrar a janela de *GUI* e tudo está funcionar tal como na versão de Windows.

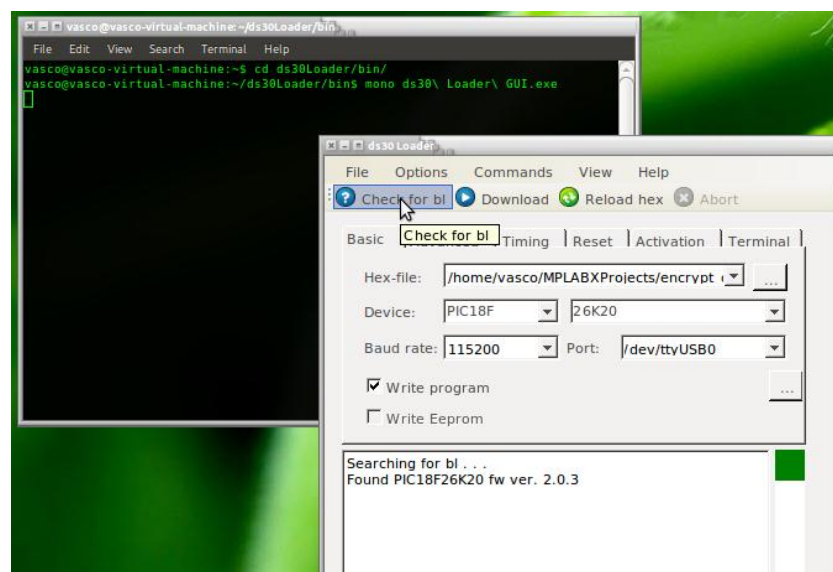


Figura 107: Janela de interface do *ds30 Loader*

Para programar deve seleccionar a porta série desejada. Para o caso de um conversor USB para SERIE, o nome da porta é "*ttyUSBx*".

Pode colocar o programa no modo avançado seleccionando "*Advanced mode*" no menu "*view*".

Notas importantes

PROBLEMA: Não consigo ver o botão "*Check for bl*", "*Reload hex*" e outras opções que tinha na versão para *Windows*.

CAUSA: Modo avançado está desactivado.

SOLUÇÃO: Active o modo avançado no menu "*View*" → "*Advanced mode*"

PROBLEMA: Quando se efectua um "*Check for bl*" na caixa de mensagens aparecem erros sem parar.

CAUSA: Problema de comunicação.

SOLUÇÃO: Por vezes esta situação acontece. Pressione "*Abort*", depois pressione o *reset* da placa e de novo pressione "*Check for bl*".

PROBLEMA: O meu conversor USB-SERIE não aparece na lista.

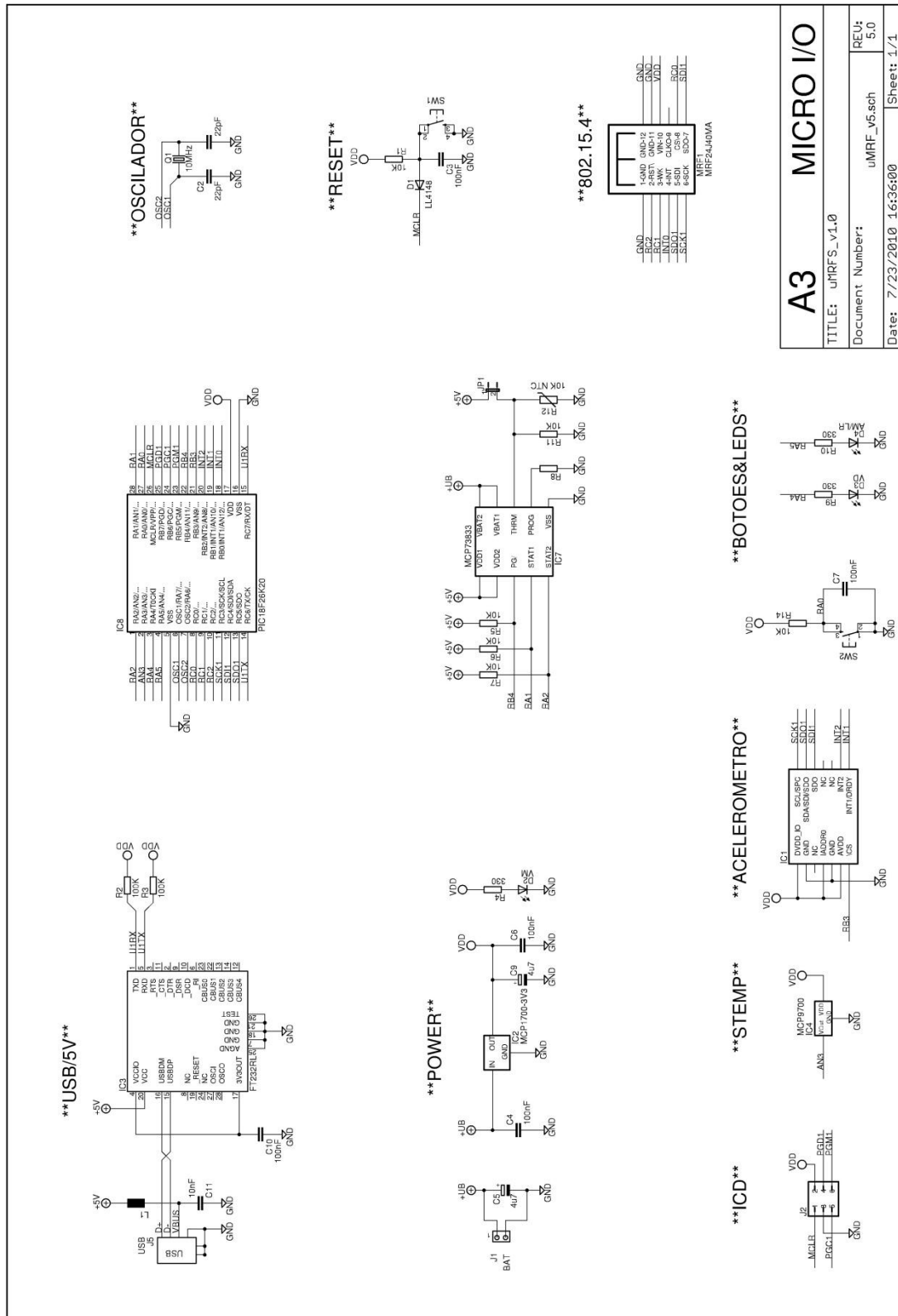
CAUSA: Possivelmente não foi detectado, correctamente instalado ou está com problemas.

SOLUÇÃO: Reinicie o *ds30 Loader*, caso não resulte faça a verificação descrita em cima relativamente aos conteúdos da pasta */dev*. Se não existir qualquer alteração nessa pasta após ligar o conversor então pode existir algum problema com o conversor, com o cabo ou com os *drivers* de USB no *Linux*. Reinicie o sistema com o conversor ligado desde o arranque e volte a tentar.

Referências relativas ao Anexo 5

- [1] MPLAB X. (2011, Março) Home Page. [Online]. http://ww1.microchip.com/downloads/mplab/X_Beta/index.html
- [2] How to Install MPLAB X on Linux (bin). (2011, Abril) Microchip Developer's Help Center. [Online]. <http://microchip.wikidot.com/mplab:install-on-linux-bin>
- [3] How to Install JAVA on Linux. (2011, Abril) Microchip Developer's Help Center. [Online]. <http://microchip.wikidot.com/mplab:how-to-install-java-on-linux>
- [4] How to swap USB drivers between MPLAB 8 and MPLAB X. (2011, Março) Microchip Developer's Help Center. [Online]. <http://microchip.wikidot.com/mplab:how-to-swap-usb-drivers-between-mplab-8-and-mplab-x>
- [5] ds30Loader. (2011, Março) Home Page. [Online]. <http://mrmackey.no-ip.org/elektronik/ds30loader/index.php>
- [6] Mono. (2011, Abril) Download Page. [Online]. <http://www.go-mono.com/mono-downloads/download.html>
- [7] Badgerports Ubuntu repositories. (2011, Abril) Home Page. [Online]. <http://badgerports.org/>

Anexo 6: Esquema da placa de testes



Anexo 7: Terminal de ligação da placa de controlo

Tabela 39: Sinais existentes no conector da placa de controlo

Pinos do conector da placa de controlo	
VCC	Alimentação positiva
GND	Alimentação negativa
/RST	Reset / Programação ICSP: MCLR
SCLK	Clock de SPI
MOSI	Linha série de saída SPI
MISO	Linha série de entrada SPI
/CS	Chip select de SPI
RX	Linha série de entrada RS232 TTL
TX	Linha série de saída RS232 TTL
IN1	Entrada 1 / Programação ICSP: PGD1
IN2	Entrada 2 / Programação ICSP: PGC1
IN3	Entrada 3 / Programação ICSP: PGM1
OUT1	Saída 1
OUT2	Saída 2
OUT3	Saída 3
FUNC	Detecção de funcionalidade
INT	Entrada de interrupção externa
AN	Entrada analógica

